Capitolul 2
Logica clasică a propoziţiilor ($\mathcal{L}_p$)

Ceea ce numim astăzi „logica clasică a propoziţiilor” ($\mathcal{L}_p$) reprezintă „numitorul comun” al unor perspective diferite de expunere formală a acestui compartiment al logicii simbolice. Capitolul de faţă analizează câteva din aceste moduri de explicitare ale $\mathcal{L}_p$: *Teoria funcţiilor de adevăr* (2.1), *Tablourile analitice* (2.2), *Axiomatica* (2.3), *Deducţia naturală* (2.4), *Calculul secvenţilor* (2.5) şi *Rezoluţia* (2.6).

2.1. Teoria funcţiilor de adevăr

2.1.1. Sintaxa $\mathcal{L}_p$

Logica propoziţiilor reprezintă partea cea mai simplă a logicii simbolice. Dacă în silogistica tradiţională, aşa cum am văzut în capitolul anterior, ne-am interesat şi de structura internă a unei propoziţii (i.e. structura subiect-predicat), în $\mathcal{L}_p$ această analiză rămâne în afara oricăror consideraţii. Descompunerea unei propoziţii o vom face până la obţinerea unor propoziţii elementare şi nimic mai mult. Dacă, de exemplu, descompunem propoziţia (compusă!) „Studenţii sunt preocupaţi de logică şi vor obţine rezultate bune” vom obţine propoziţiile elementare „Studenţii sunt preocupaţi de logică” şi „Studenţii vor obţine note bune”, fără să mai avem în vedere structura internă a acestor propoziţii elementare. Aşadar, unităţile sintactice nedecompozabile ale $\mathcal{L}_p$ sunt propoziţiile elementare. Din propoziţii elementare, aidoma exemplului de mai sus, putem construi propoziţii compuse de varii feluri. Date fiind propoziţiile elementare „Clujul este un oraş transilvan” şi „A venit toamna” putem obţine „*Nu* a venit toamna”, „Clujul este un oraş transilvan *sau* a venit toamna”, „*Dacă* Clujul este un oraş transilvan, *atunci* a venit toamna” etc. Aşadar, asupra propoziţiilor putem *opera* în diverse moduri. În primul caz am *negat* propoziţia dată, în al doilea am *conectat disjunctiv* cele două propoziţii, iar în cel de-al treilea le-am *conectat implicativ* etc.

Logica *simbolică* a propoziţiilor nu se interesează însă de conţinutul propoziţiilor (i.e. de ceea ce ele exprimă), ci de relaţiile logice posibile dintre ele. Motiv pentru care recurge la o simbolizare corespunzătoare şi, implicit, la un nou nivel de abstractizare. Pe scurt, limbajul logicii propoziţiilor, $\mathcal{L}_p$ [1], conţine următoarele categorii de simboluri (i.e. alfabetul):

1. Simboluri pentru variabile propoziţionale: $p, q, r, ..., p_1, p_2, ..., q_1, q_2, ...$
2. Simboluri pentru operatorii logici: $\neg, \wedge, \vee, \supset, ...$
3. Simboluri pentru constante: 1 şi 0.

[1] „$\mathcal{L}_p$” înseamnă logica simbolică a propoziţiilor, aşa cum este ea explicitată în limbajul expus.

4. Simboluri auxiliare:), (,], [, }, {.

Simbolurile *p*, *q*, *r*,... denotă aşadar propoziţii *arbitrare*; $\neg$ - este operatorul unar al negaţiei; $\wedge, \vee, \supset$ sunt, corespunzător, operatori *binari*: comjuncţie, disjuncţie, implicaţie[2]; 1 şi 0 denotă valorile logice „adevărat" şi „fals"[3].

Formulă a $\mathcal{L}_p$. Acest concept va fi definit pe baza unor reguli recursive, reguli care permit obţinerea de noi formule din cele deja construite. Prin $\alpha, \beta, \gamma, ..., \alpha_1, \alpha_2 ...$ înţelegem, în cele ce urmează, formule arbitrare ale $\mathcal{L}_p$.

Definiţie 1. *a) Orice variabilă propoziţională este o formulă a* $\mathcal{L}_p$.

b) Dacă α *este o formulă a* $\mathcal{L}_p$, *atunci* $\neg\alpha$ *este o formulă a* $\mathcal{L}_p$.

c) Dacă α *şi* β *sunt formule ale* $\mathcal{L}_p$, *atunci* $\alpha \circ \beta$ *este o formulă a* $\mathcal{L}_p$

(*unde „* $\circ$ *" denotă oricare din următorii 10 operatori binari:* $\wedge, \vee, \supset, \subset, \equiv, /, \downarrow, +, \not\subset, \#$[4]).

Prin $\Gamma, \Delta, \Lambda, ...$ *vom înţelege mulţimi de formule ale* $\mathcal{L}_p$.

Definiţie 2. Gradul unei formule este numărul ocurenţelor operatorilor logici din formulă, i.e.:

a) Orice variabilă propoziţională are gradul zero.

b) Dacă α *are gradul n, atunci* $\neg\alpha$ *are gradul* $n+1$.

c) *Dacă* α *şi* β *au gradele n, respectiv m, atunci* $\alpha \circ \beta$ *are gradul* $n+m+1$.

2.1.2. Semantica $\mathcal{L}_p$

2.1.2.1. Funcţii de adevăr

Pentru a decupa semnificaţia integrală a unui operator logic nu este suficient nicidecum să ne rezumăm la exemple din limbajul natural. Va trebui ca înţelesul lor să fie complet explicitat prin definiţii semantice. Aceste definiţii arată, în fiecare caz în parte, modul în care valoarea logică a propoziţiei compuse (i.e. a propoziţiei care conţine operatorul respectiv) depinde strict de valorile logice ale propoziţiilor care-o compun. Să dăm câteva exemple. Fie propoziţia „Stiloul este albastru". Considerăm că propoziţia este adevărată. Dacă negăm această propoziţie, atunci vom obţine propoziţia compusă „Nu este adevărat că stiloul este albastru" sau „Stiloul nu este albastru", simbolic $\neg p$, şi care este o propoziţie falsă. Invers, dacă p era falsă, atunci $\neg p$ este adevărată. Corelăm aceste valori logice ale propoziţiei compuse, $\neg p$, cu valorile logice ale componentei ei, *p*, într-un tabel sau *matrice* şi obţinem:

p	$\neg p$
1	0
0	1

Aici *p* este argumentul funcţiei negaţie.

Această matrice reprezintă definiţia semantică a negaţiei.

---

[2] O expunere în detaliu a operatorilor binari o facem în § *Semantica* $\mathcal{L}_p$.

[3] Singurele cu care operăm, logica clasică fiind o logică bivalentă.

[4] Din motive tipografice simbolul nonimplicaţiei a putut fi redat doar prin #.

La fel putem proceda considerând *p*: „2 + 2 = 4” şi *q*: „Tabla este neagră”. Dând valori logice lui *p* şi *q*, în toate combinaţiile posibile, putem construi definiţii pentru toţi operatorii binari ai $\mathcal{L}_p$. Să menţionăm câţiva:

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

;

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

;

p	q	$p \supset q$
1	1	1
1	0	0
0	1	1
0	0	1

;

p	q	$p \equiv q$
1	1	1
1	0	0
0	1	0
0	0	1

p şi *q* sunt argumentele funcţiilor corespunzătoare.

Cunoscând matricea unui operator, putem construi, corespunzător, definiţia acestuia. Conjuncţia, de exemplu, este funcţia de adevăr care ia valoarea logică adevărat atunci când ambele argumente sunt adevărate. Disjuncţia este funcţia de adevăr care ia valoarea logică adevărat dacă cel puţin un argument este adevărat. Argumentele implicaţiei ($\supset$) se numesc *p* = antecedent şi *q* = consecvent. Din matricea corespunzătoare rezultă că implicaţia este falsă doar dacă antecedentul este adevărat şi consecventul fals. În fine, echivalenţa ($\equiv$) ia valoarea logică adevărat doar dacă argumentele *p, q* au aceeaşi valoare logică. Trebuie să remarcăm însă următorul fapt: ceea ce intră în discuţie în stabilirea valorii de adevăr a propoziţiei compuse (i.e. în definirea funcţiei/operatorului) este *strict* valoarea logică a propoziţiilor componente şi nimic altceva. În caz contrar, propoziţia implicativă, construită cu *p* şi *q*, de mai sus ar fi o propoziţie deloc uzuală în înţelegerea curentă, de genul: „Dacă 2 + 2 = 4, atunci tabla este neagră”. Aşadar, orice consideraţie cu privire la sensul propoziţiilor este eliminată. Acesta este motivul pentru care aceste funcţii se numesc *funcţii de adevăr*.

Definiţie 1. Funcţia de adevăr este o funcţie de la mulţimea valorilor logice ale propoziţiilor componente la valoarea logică a propoziţiei compuse. (*simbolic:* $f_A : A(p_1,..., p_n) \rightarrow A_\varphi(p_1,..., p_n)$*, unde* φ *este un operator logic*).

Numărul funcţiilor care pot fi construite în logica bivalentă depinde de numărul variabilelor propoziţionale, *n*, şi se calculează după relaţia simplă $\mathrm{Nr}\varphi = 2^{(2^n)}$.

Pentru *n* = 1, numărul funcţiilor este de patru, adică:

p	$\varphi_1(p)$	$\varphi_2(p)$	$\varphi_3(p)$	$\varphi_4(p)$
1	1	1	0	0
0	1	0	1	0

Tabelul de adevăr de mai sus are cinci coloane şi două linii. Prima coloană conţine asignările făcute variabilei propoziţionale *p*, iar celelalte patru reprezintă funcţiile *unare* corespunzătoare, adică: *verum* (*p*), *p*, $\neg p$ şi *falsum* (*p*).

Definiţie 2. Prin interpretarea unei variabile propoziţionale se înţelege asignarea (atribuirea) unei valori de adevăr variabilei respective.

Dacă cunoaştem tabelul de adevăr al unei funcţii, atunci cunoaştem semnificaţia funcţiei, respectiv îi putem construi definiţia. Din cele patru funcţii de mai sus, $\varphi_3(p)$ este tocmai *negaţia*. Negaţia este aşadar funcţia de adevăr (unară!) care ia valoarea logică adevărat dacă p este falsă; fals în caz contrar. La fel putem construi definiţii şi pentru celelalte funcţii unare.

Pentru $n = 2$, $\mathrm{Nr}\varphi = 2^{\left(2^2\right)} = 16$. Cele 16 funcţii *binare* distincte sunt redate de ceea ce se numeşte *Tabelul lui Wittgenstein* [5].

p	q	$\varphi_1(p,q)$[6]	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		$\mathcal{T}$	$\vee$	$\subset$	p	$\supset$	q	$\equiv$	$\wedge$	/	+	$\neg q$	#	$\neg p$	$\not\subset$	$\downarrow$	C

$$L_1 = \frac{\mathrm{Nr}\varphi}{2} = 8 \quad ; \quad L_2 = \frac{\mathrm{Nr}\varphi}{2^2} = 4 \quad ; \quad L_3 = \frac{\mathrm{Nr}\varphi}{2^3} = 2 \quad ; \quad L_4 = \frac{\mathrm{Nr}\varphi}{2^4} = 1$$

Tabelul conţine 18 coloane şi 4 linii. Primele două coloane reprezintă toate categoriile de asignări făcute variabilelor p şi q, iar celelalte funcţiile binare corespunzătoare. Completarea liniilor o facem după următorul algoritm simplu: în linia întâi (L_1) vom scrie alternativ valori *omogene* al căror număr este numărul total al funcţiilor (16) împărţit la 2, pentru linia a doua vom împărţi acest număr la 2^2, apoi la 2^3 pentru linia a treia şi, în fine, la 2^4 pentru linia a patra. Cele 16 funcţii binare sunt, corespunzător: tautologia ($\mathcal{T}$), disjuncţia neexclusivă ($\vee$), implicaţia conversă ($\subset$), prependenţa (p), implicaţia ($\supset$), postpendenţa (q), echivalenţa ($\equiv$), conjuncţia ($\wedge$), incompatibilitatea (/), disjuncţia exclusivă ($+$), postnonpendenţa ($\neg q$), non-implicaţia (#), prenonpendenţa ($\neg p$), nonimplicaţia conversă ($\not\subset$), rejecţia ($\downarrow$) şi contradicţia (C).

Din construcţia tabelului se poate observa că funcţiile $\varphi_9 - \varphi_{16}$ se pot obţine, corespunzător, din negarea funcţiilor $\varphi_8 - \varphi_1$, motiv pentru care incompatibilitatea de mai sus se mai numeşte şi anticonjuncţie, disjuncţia exclusivă – antiechivalenţă, rejecţia – antidisjuncţie etc.

Tot în construcţia tabelului observăm că φ_1 este întotdeauna adevărată, φ_{16} este întotdeauna falsă, iar celelalte funcţii conţin în matricea lor valori de adevăr neomogene. Motiv pentru care putem opera următoarea clasificare a funcţiilor de adevăr: *valide* (tautologii, adevăruri logice), *nesatisfiabile* (contradicţii, falsităţi logice) şi *satisfiabile* (realizabile), terminologie pe care o vom aplica cu referire la orice formulă a $\mathcal{L}_\mathrm{p}$ (aşa cum vom arăta mai jos).

[5] Comp. L. Wittgenstein, *Tractatus logico-philosophicus*, 5.101.
[6] Din motive tipografice pentru celelalte funcţii vom omite argumentele.

2.1.2.2. Dualitatea

Definiție. Doi operatori logici[7] se numesc *duali* dacă matricea unuia se obține din matricea celuilalt înlocuind reciproc valorile logice 1 şi 0 în toate ocurențele lor.

Exemplu. Dacă în matricea operatorului $\equiv$ executăm aceste înlocuiri obținem matricea operatorului $+$.

p	q	$p \equiv q$
1	1	1
1	0	0
0	1	0
0	0	1

p	q	$p + q$
0	0	0
0	1	1
1	0	1
1	1	0

Dacă şi în celelalte cazuri procedăm similar, vom găsi următoarele perechi de operatori duali: $\top - \bot$, $\vee - \wedge$, $\supset - \not\subset$, $\subset - \#$ (exercițiu).

Ceilalți, $p, q, \neg p$ şi $\neg q$, constituie gama operatorilor autoduali. Operatorul unar $\neg$ este de asemenea autodual (de ce?).

2.1.2.3. Procedeul matriceal ca procedeu de decizie în $\mathcal{L}_p$

Definiție 1. O formulă α a $\mathcal{L}_p$ se numeşte validă (simbolic: $\vDash \alpha$) dacă pentru orice interpretare a variabilelor sale α ia valoarea logică adevărat.

Definiție 2. O formulă α a $\mathcal{L}_p$ se numeşte nesatisfiabilă dacă pentru orice interpretare a variabilelor sale α ia valoarea logică fals.

Definiție 3. O formulă α a $\mathcal{L}_p$ se numeşte satisfiabilă dacă există două interpretări ale variabilelor ei în care α ia valori logice opuse.

Remarcă. În sens *larg* satisfiabilitatea nu exclude validitatea, α fiind satisfiabilă dacă există (cel puțin) o interpretare în care α este adevărată.

Pentru a constata validitatea unei formule (valide!) procedeul matriceal este unul foarte practic, atunci când numărul variabilelor formulei nu este prea mare (mai mare de 3).

Să luăm un exemplu:

$\alpha^{8}(p,q,r)$: $[(p /_{1} q) \supset_{3}^{2} \neg r] \wedge_{6} (q \vee_{5}^{4} \neg p)$

Numărul liniilor matricei este 2^n, unde n reprezintă numărul variabilelor propoziționale. Numărul coloanelor este dat de numărul operatorilor logici ai formulei plus numărul variabilelor propoziționale. Vom construi aşadar o matrice (tabel de adevăr) cu 8 linii

[7] Corespunzător, funcții de adevăr, în cele ce urmează vom considera cele două expresii interschimbabile.

[8] Menționarea variabilelor, după simbolul „α”, nu este necesară.

şi 9 coloane. Pentru a facilita construcția matricei, putem nota operatorii logici în ordinea executării operațiilor.

p	q	r	$/_1$	$\neg_2$	$\supset_3$	$\neg_4$	$\vee_5$	$\wedge_6$
1	1	1	0	0	1	0	1	1
1	1	0	0	1	1	0	1	1
1	0	1	1	0	0	0	0	0
1	0	0	1	1	1	0	0	0
0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	1	1
0	0	1	1	0	0	1	1	0
0	0	0	1	1	1	1	1	1

Completarea interpretărilor variabilelor o facem şi în acest caz după algoritmul mai sus aplicat (în cazul Tabelului lui Wittgenstein), numai că aici avem, corespunzător, coloane, nu linii.

Coloana ultimei operații menționate, $\wedge_6$, reprezintă valorile logice ale formulei α, pentru toate cele 8 interpretări ale variabilelor sale. Aşa cum ne arată matricea, avem o funcție realizabilă.

Să mai luăm un exemplu. Fie $\beta : [(p \wedge \neg r)/ q] \supset [(\neg p \downarrow \neg q) \supset p]$.

Facem, corespunzător, matricea, fără a mai numerota operatorii logici.

p	q	r	$\neg r$	$p \wedge \neg r$	$(p \wedge \neg r)/ q$	$\neg p$	$\neg q$	$\neg p \downarrow \neg q$	$(\neg p \downarrow \neg q) \supset p$	β
1	1	1	0	0	1	0	0	1	1	1
1	1	0	1	1	0	0	0	1	1	1
1	0	1	0	0	1	0	1	0	1	1
1	0	0	1	1	1	0	1	0	1	1
0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	1	0	0	1	1
0	0	1	0	0	1	1	1	0	1	1
0	0	0	1	0	1	1	1	0	1	1

Cum coloana finală a matricei formulei β conține o serie omogenă de valori de 1, rezultă că β este o formulă validă.

Formule valide remarcabile ale $\mathcal{L}_p$

Prezentăm mai jos câteva tautologii remarcabile ale $\mathcal{L}_p$, cu referire la operatorii menționați în frontul fiecărei grupe $(\neg,\wedge,\vee,\supset,\equiv)$. În dreptul fiecărei formule vom specifica denumirea uzuală a tautologiei respective în $\mathcal{L}_p$.

$(\neg)$ 1. $\neg...\neg^{2k} p \equiv p$; $k = 0,1,2,...$; 1 şi 2 Eliminarea negațiilor multiple

2. $\neg...\neg^{2k+1} p \equiv \neg p$; $k = 0,1,2,...$

$(\wedge)$ 3. $(p \wedge p) \equiv p$; idempotența conjuncției

4. $(p \wedge q) \equiv (q \wedge p)$; comutativitatea conjuncției

5. $[p \wedge (q \wedge r)] \equiv [(p \wedge q) \wedge r]$; asociativitatea conjuncției

6. $[p \wedge (q \vee r)] \equiv [(p \wedge q) \vee (p \wedge r)]$; distributivitatea conjuncției (în raport cu disjuncția)

7. $[p \wedge (p \vee q)] \equiv p$; 7 – 9 Absorbția conjuncției

8. $[p \wedge (p \wedge q)] \equiv (p \wedge q)$

9. $[p \wedge (q \vee \neg q)] \equiv p$

10. $(p \wedge q) \supset p$; 10 – 13 Atenuarea conjuncției

11. $(p \wedge q) \supset q$

12. $(p \wedge q) \supset (p \vee q)$

13. $(p \wedge q) \supset (p \supset q)$

14. $(p \wedge q) \equiv \neg(\neg p \vee \neg q)$; 14 şi 15 Legile lui De Morgan

15. $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$

16. $\neg(p \wedge \neg p)$; principiul noncontradicției

$(\vee)$ 17. $(p \vee p) \equiv p$; idempotența disjuncției

18. $(p \vee q) \equiv (q \vee p)$; comutativiatea disjuncției

19. $[p \vee (q \vee r)] \equiv [(p \vee q) \vee r]$; asociativitatea disjuncției

20. $[p \vee (q \wedge r)] \equiv [(p \vee q) \wedge (p \vee r)]$; distributivitatea disjuncției (în raport cu conjuncția)

21. $[p \vee (p \vee q)] \equiv (p \vee q)$; 21 – 23 Absorbția disjuncției

22. $[p \vee (p \wedge q)] \equiv p$

23. $[p \vee (q \wedge \neg q)] \equiv p$

24. $p \supset (p \vee q)$; 24 şi 25 Introducerea disjuncției

25. $q \supset (p \vee q)$

26. $(p \vee q) \equiv \neg(\neg p \wedge \neg q)$; 26 şi 27 Legile lui De Morgan

27. $\neg(p \vee q) \equiv (\neg p \wedge \neg q)$

28. $p \vee \neg p$; principiul terțului exclus

$(\supset)$ 29. $p \supset p$; reflexivitatea implicației

30. $[(p \supset q) \wedge (q \supset r)] \supset (p \supset r)$; tranzitivitatea implicației

31. $[p \supset (q \supset r)] \equiv [(p \wedge q) \supset r]$; legea importației / exportației

32. $[p \supset (q \supset r)] \equiv [q \supset (p \supset r)]$; permutarea premiselor
33. $[p \supset (p \supset q)] \equiv (p \supset q)$; absorbția implicației
34. $[p \supset (q \supset r)] \equiv [(p \supset q) \supset (p \supset r)]$; autodistributivitatea implicației
35. $(p \supset q) \equiv (\neg q \supset \neg p)$; 35 – 37 Legi ale contrapoziției
36. $(\neg p \supset q) \equiv (\neg q \supset p)$
37. $(p \supset \neg q) \equiv (q \supset \neg p)$
38. $p \supset (q \supset p)$; 38 şi 39 Paradoxurile implicației
39. $\neg p \supset (p \supset q)$
40. $p \supset (\neg p \supset q)$; ex falso quodlibet
41. $(p \supset q) \supset [(p \wedge r) \supset q]$; 41 – 44 atenuarea implicației
42. $(p \supset q) \supset [(r \supset p) \supset (r \supset q)]$
43. $(p \supset q) \supset [(q \supset r) \supset (p \supset r)]$
44. $(p \supset q) \supset [(p \vee r) \supset (q \vee r)]$
45. $[p \wedge (p \supset q)] \supset q$; modus ponens
46. $[p \supset (p \supset q)] \supset q$; modus ponendo ponens
47. $[p \supset (q \supset \neg p)] \supset \neg q$; 47 şi 48 modus tollendo tollens
48. $[(p \supset q) \wedge \neg q] \supset \neg p$
49. $[(p \supset q) \wedge (\neg p \supset q)] \supset q$; dilema constructivă
50. $(p \supset \neg p) \supset \neg p$; 50 – 52 reductio ad absurdum
51. $(\neg p \supset p) \supset p$
52. $[(p \supset q) \wedge (p \supset \neg q)] \supset \neg p$
53. $[(p \supset q) \supset p] \supset p$; legea lui Peirce

($\equiv$) 54. $p \equiv p$; reflexivitatea echivalenței
55. $(p \equiv q) \supset (q \equiv p)$; simetria echivalenței
56. $[(p \equiv q) \wedge (q \equiv r)] \supset (p \equiv r)$; tranzitivitatea echivalenței
57. $[p \equiv (q \equiv r)] \equiv [(p \equiv q) \equiv r]$; asociativitatea echivalenței
58. $(p \equiv q) \equiv (\neg p \equiv \neg q)$; 58 şi 59 contrapoziția echivalenței
59. $(\neg p \equiv q) \equiv (p \equiv \neg q)$
60. $(p \equiv q) \supset (p \supset q)$; 60 şi 61 atenuarea echivalenței
61. $(p \equiv q) \supset (q \supset p)$
62. $\neg(p \equiv q) \equiv (\neg p \equiv q)$; respingerea echivalenței
63. $(p \equiv q) \equiv [(p \wedge q) \vee (\neg p \wedge \neg q)]$; condițiile de adevăr ale echivalenței

2.1.2.4. Relația de consecință semantică

Pentru explicitarea înțelesului acestui concept să dăm mai întâi un exemplu. Fie următoarele formule ale $\mathcal{L}_p$: $\alpha:(p\equiv q)\downarrow(p\wedge q)$, $\beta:(p\wedge q)\supset\neg r$ şi $\gamma:p\vee q$. Să prezentăm în tabelul de mai jos matricea celor trei formule.

p	q	r	$p\equiv q$	$p\wedge q$	α	$\neg r$	β	γ
1	1	1	1	1	0	0	0	1
1	1	0	1	1	0	1	1	1
1	0	1	0	0	1	0	1	1
1	0	0	0	0	1	1	1	1
0	1	1	0	0	1	0	1	1
0	1	0	0	0	1	1	1	1
0	0	1	1	0	0	0	1	0
0	0	0	1	0	0	1	1	0

Pe baza acestei matrice putem observa că de ori câte ori formulele α şi β sunt *simultan* adevărate şi formula γ este adevărată (conversa nefiind valabilă). În acest caz vom spune că formula γ este o *consecință semantică* a formulelor α şi β. Mai putem constata că de ori câte ori α este adevărată şi β este adevărată (reciproca nefiind nici în acest caz valabilă). Şi în acest caz vom spune că β este consecință semantică a lui α.

Definiție. Fie $\alpha_1,...,\alpha_n,\beta$ formule ale $\mathcal{L}_p$. β este o consecință semantică a formulelor $\alpha_1,...,\alpha_n$ (simbolic $\alpha_1,...,\alpha_n\models\beta$) dacă pentru orice interpretare a variabilelor propoziționale care apar în $\alpha_1,...,\alpha_n,\beta$, formula β este adevărată de ori câte ori $\alpha_1,...,\alpha_n$ sunt simultan adevărate ($\alpha_1,...,\alpha_n$ sunt premisele iar β este consecința lor semantică).

Exerciții.

1. Se dau următoarele formule ale $\mathcal{L}_p$: $\alpha_1:(p\equiv q)\supset(p\supset q)$, $\alpha_2:(p\wedge q)\supset(p\supset q)$, $\alpha_3:(p\vee q)\supset(p\wedge q)$, $\alpha_4:(p\supset q)\supset[(p\wedge r)\supset q]$. Stabiliți cazurile în care relația de consecință semantică are loc.
2. Ce putem spune despre relația $\alpha\models\beta$ dacă β este o formulă validă a $\mathcal{L}_p$? Dar dacă β este o formulă nesatisfiabilă?
3. Construiți o demonstrație pentru următoarele enunțuri:
   a) Dacă $\models\alpha$ şi $\models\alpha\supset\beta$, atunci $\models\beta$.
   b) Dacă $\models\alpha$ şi $\models\alpha\equiv\beta$, atunci $\models\beta$.
4. Argumentați că următorul condițional are loc:
   Pentru $n\geq 1$: $\alpha_1,...,\alpha_n\models\beta$ ddacă $\alpha_1\wedge...\wedge\alpha_n\models\beta$.

2.1.2.5. Scheme deductive cu operatori ai $\mathcal{L}_p$

Din cele 16 funcții de adevăr binare, 10 pot figura în scheme deductive valide. Celelalte 6 (p, q, $\neg p$, $\neg q$, $\mathcal{T}$, $\mathcal{C}$), numite şi funcții *degenerate*, nu exprimă propriu-zis corelații logice între argumentele lor, motiv pentru care nu pot construi scheme deductive valide. Să considerăm în continuare câteva exemple de scheme valide.

Cu ajutorul implicației materiale putem construi următoarele scheme:

$$\frac{p \supset q \quad p}{q}$$ modus ponens

$$\frac{p \supset q \quad \neg q}{\neg p}$$ modus tollens

p	q	$p \supset q$
1	1	1
1	0	0
0	1	1
0	0	1

Verificarea validității lor o putem face fie construind o formulă implicativă de genul premise $\supset$ concluzie, pe care-o verificăm cu ajutorul procedeului matriceal, adică $[(p \supset q) \wedge p] \supset q$, fie o justificăm pe baza matricei (i.e. definiției semantice a) implicației. În acest din urmă caz, alegem din matrice liniile care stipulează condițiile din premisele schemei, adică $p = 1$ şi $(p \supset q) = 1$, respectiv doar prima linie. Pentru concluzie ne rămâne o singură alegere, $q = 1$. La fel putem justifica şi *modus tollens*.

În schimb, următoarele scheme sunt nevalide (de ce?):

$$\frac{p \supset q \quad \neg p}{q} \qquad \frac{p \supset q \quad q}{p}$$

Cu *incompatibilitatea* putem construi alte scheme deductive:

$$\frac{p \,/\, q \quad p}{\neg q} \quad \text{sau} \quad \frac{p \,/\, q \quad q}{\neg p}$$

Exerciții.

1. Se dau următoarele definiții:

Def. 1. O formulă a $\mathcal{L}_p$ este *satisfiabilă* dacă există două interpretări distincte ale variabilelor sale astfel încât formula ia, corespunzător, valori logice distincte.

Def. 2. O formulă a $\mathcal{L}_p$ este *satisfiabilă* dacă există cel puțin o interpretare în care formula este adevărată.

Def. 3 O formulă a $\mathcal{L}_p$ este *satisfiabilă* dacă formula nu este nesatisfiabilă.

Ce raporturi logice putem stabili între cele trei definiții?

2. Unele funcții din tabelul lui Wittgenstein exprimă *raporturi de condiționare* ($\supset, \subset, \equiv, \wedge, \downarrow$ şi $\mathcal{T}$), altele exprimă *raporturi de opoziție* (negațiile funcțiilor precedente) iar altele exprimă *raporturi mixte* (p, q, $\neg p$ şi $\neg q$).

a) Arătați că funcțiile p, q, $\neg p$ şi $\neg q$ exprimă raporturi mixte (i.e. pot fi redate, fiecare, ca o conjuncție dintre o funcție care exprimă un raport de condiționare şi una care exprimă un raport de opoziție).

b) Verificați, prin transformări sintactice adecvate, că relațiile respective au loc.

3. Construiți toate schemele valide în care figurează operatori binari ai $\mathcal{L}_p$. Argumentați validitatea lor.

4. Formulați concluzia, în termenii incompatibilității, pe baza următoarelor premise:

$(p/q)/[(p/p) / (q/q)]$

q/q

5. Construiți două scheme deductive valide, una în care una dintre premise este $\neg p \supset q$, iar cealaltă în care una din premise este $p \supset \neg q$.

6. Formulați concluzia, în termenii rejecției, pe baza următoarelor premise:

$(p \wedge q) \downarrow (p \wedge q)$

$(p / p) / (p / p)$

7. Explicați simbolic ideea excluderii covalenței propozițiilor.

8. Să se determine, în termenii rejecției, cea de-a doua premisă împreună cu care expresia de mai jos poate constitui o schemă deductivă validă. Formulați concluzia.

$(p / p) / (q / q)$

9. Constituie următoarea combinație de premise o schemă deductivă validă? De ce?

p / p

$p / (q / q)$

10. Ce concluzie se poate formula, în termenii rejecției, pe baza următoarelor premise:

$[(p \downarrow p) \downarrow q] \downarrow [(p \downarrow p) \downarrow q]$

$(p \downarrow p) \downarrow (p \downarrow p)$

11. Construiți o schemă deductivă validă în care premisele sunt

$(p \supset q) \supset \neg(q \supset p)$

p / p

2.1.3. Transformări echiveridice în logica propozițiilor

2.1.3.1. Conceptul echiveridicității

Vom ilustra mai întâi ideea de echiveridicitate, pe baza câtorva exemple simple.
a) Fie următoarele două formule ale $\mathcal{L}_p$:
$\alpha : p \wedge (q \vee r)$; $\beta : (p \wedge q) \vee (p \wedge r)$

Să vedem în ce relație logică se află ele, comparând matricele:

p	q	r	$q \vee r$	$p \wedge (q \vee r)$	$p \wedge q$	$p \wedge r$	$(p \wedge q) \vee (p \wedge r)$	$\alpha \equiv \beta$
1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0	1
0	1	1	1	0	0	0	0	1
0	1	0	1	0	0	0	0	1
0	0	1	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1

Vom constata că valorile din coloana finală (respectiv valorile acestor funcții de adevăr pentru asignările corespunzătoare) sunt identice. Vom spune că α şi β sunt formule *echiveridice*.

b) Fie acum formulele $\alpha : p \supset q$ şi $\beta : \neg p \vee q$. Similar,

p	q	$p \supset q$	$\neg p$	$\neg p \vee q$	$\alpha \equiv \beta$
1	1	1	0	1	1
1	0	0	0	0	1
0	1	1	1	1	1
0	0	1	1	1	1

Şi în acest caz este valabilă constatarea de mai sus.

c) În fine, considerăm formulele: $\alpha : [(p/q) + \neg r] \supset [(p \wedge \neg q) \supset \neg q]$ şi $\beta : \neg(s \vee \neg s) \supset [(p \downarrow q) \wedge \neg t]$.

Fără să facem matricele corespunzătoare putem constata că ambele formule sunt valide: prima este o implicație cu consecventul adevărat iar a doua o implicație cu antecedentul fals. Şi în acest caz cele două formule sunt echiveridice.

Să construim acum o definiție pentru conceptul echiveridicității. Fie $p_1, ..., p_n$ $(n \geq 1)$ toate variabilele care apar într-o formulă α, fie $q_1, ..., q_m$ $(m \geq 1)$ toate variabilele care apar într-o formulă β.

Definiție. Formulele α şi β se numesc echiveridice (sau semantic echivalente) dacă pentru orice interpretare dată variabilelor $p_1,...,q_m$ ele iau aceeaşi valoare logică (simbolic α eq β).

Remarcă 1. Pentru a fi echiveridice formulele nu trebuie să conțină neaparat aceiaşi operatori logici (comp. exemplul b) sau aceleaşi variabile (sintactic sau numeric) (comp. exemplul c).

Remarcă 2. Echiveridicitatea nu trebuie confundată cu operatorul echivalență, „≡", din Tabelul lui Wittgenstein. Aceasta este un operator binar al limbajului obiect care, pus între două formule, α şi β, formează o nouă formulă mai complexă $\alpha \equiv \beta$ a limbajului obiect. „Eq" este, în schimb, un operator binar al metalimbajului care, pus între simbolurile α şi β, generează expresia „α *eq* β", în care α şi β sunt termenii relației „eq" şi nu formule ca atare (α înseamnă „formula α", iar β „formula β"). Fireşte, aşa cum am constatat din exemplele de mai sus, între „eq" şi „≡" există următoarea relație:

α eq β ddacă $\alpha \equiv \beta$ este o formulă validă (i.e. $\vDash \alpha \equiv \beta$).

Aşadar, două formule α şi β ale $\mathcal{L}_p$ sunt echiveridice ddacă ele sunt logic echivalente, adică pentru orice interpretare a variabilelor lor ele iau aceeaşi valoare logică şi deci $\vDash \alpha \equiv \beta$.

Echiveridicitatea are următoarele proprietăți:

1. *Refl*: α eq α.
2. *Sym*: Dacă α eq β, atunci β eq α.
3. *Trans*: Dacă α eq β şi β eq γ atunci α eq γ.

2.1.3.2. Operatori fundamentali, operatori derivați

Definiție 1. Un operator logic introdus cu ajutorul unei definiții semantice (matrice) se numeşte operator fundamental (la fel putem spune, corespunzător, despre o funcție de adevăr).

Definiție 2. Un operator logic introdus cu ajutorul unei definiții sintactice se numeşte operator derivat (respectiv o funcție de adevăr derivată).

Exemple: $p \supset q =_{df} \neg p \vee q$; $p / q =_{df} \neg p \vee \neg q$ etc.

Fie, în cele ce urmează, o mulțime $M = \{O_1,...,O_n\}$, $n \geq 1$, de operatori fundamentali.

Definiție 3. Un operator O_i se numeşte funcțional independent de ceilalți operatori din mulțimea M dacă el nu se poate defini sintactic prin ceilalți operatori.

Definiție 4. O mulțime de operatori fundamentali $M = \{O_1,...,O_n\}$, $n \geq 1$, se numeşte independentă dacă fiecare operator este funcțional independent de ceilalți operatori.

Exemple.

1. Fie $M = \{\neg, \vee, \wedge\}$. M nu formează o mulțime independentă, deoarece $\wedge$ se poate defini sintactic prin $\neg$ şi $\vee$:

$$p \wedge q =_{df} \neg(\neg p \vee \neg q)$$

Similar, $\vee$ se poate defini prin $\wedge$ şi $\neg$.

2. $M = \{\neg, \wedge\}$ este independentă.

Argument. Trebuie să arătăm că operatorul $\wedge$ nu poate fi definit doar cu operatorul $\neg$ şi invers. Din variabile şi din negație putem construi doar formule de forma $p, \neg p, \neg\neg p, \neg...\neg p$ etc. Aceste formule sunt fie echiveridice cu p, fie cu $\neg p$, pe baza formulelor 1 şi 2, § 2.3.

Cu ajutorul negației se pot construi aşadar numai acele funcții binare care sunt echiveridice fie cu p, fie cu q, fie cu $\neg p$, fie cu $\neg q$, adică:

p	q	$\varphi_1(p,q)$	$\varphi_2(p,q)$	$\varphi_3(p,q)$	$\varphi_4(p,q)$
1	1	1	1	0	0
1	0	1	0	0	1
0	1	0	1	1	0
0	0	0	0	1	1

Întrucât nici una din aceste funcții nu este echiveridică cu conjuncția, conchidem că operatorul $\wedge$ nu poate fi definit doar prin intermediul negației. Similar, $\neg$ nu poate fi definită doar cu ajutorul $\wedge$. Cu ajutorul $\wedge$ nu pot fi construite decât formule de forma $p \wedge p, p \wedge ... \wedge p$, toate fiind echiveridice cu p.

2.1.3.3.Completitudine funcțională

Definiția 3.3. O mulțime M de operatori fundamentali se numeşte funcțional completă dacă orice alt operator al $\mathcal{L}_p$ *se poate defini prin operatorii mulțimii M.*

Vom da în continuare câteva exemple de mulțimi funcțional complete de operatori logici: $M_1 = \{\neg, \wedge\}$, $M_2 = \{\neg, \vee\}$, $M_3 = \{\neg, \supset\}$, $M_4 = \{/\}$ şi $M_5 = \{\downarrow\}$. Vom considera în cele ce urmează doar următorii operatori din tabelul lui Wittgenstein: $\wedge, \vee, \supset, \equiv, +, /, \downarrow$.

$M_1 = \{\neg, \wedge\}$ este funcțional completă

$p \vee q$ eq $\neg(\neg p \wedge \neg q)$

$p \supset q$ eq $\neg(p \wedge \neg q)$

$p \equiv q$ eq $\neg(p \wedge \neg q) \wedge (q \wedge \neg p)$

$p + q$ eq $\neg[\neg(p \wedge \neg q) \wedge \neg(q \wedge \neg p)]$

p / q eq $\neg(p \wedge q)$

$p \downarrow q$ eq $\neg p \wedge \neg q$

$M_2 = \{\neg, \vee\}$ este funcțional completă

$p \wedge q$ eq $\neg(\neg p \vee \neg q)$

$p \supset q$ eq $\neg p \vee q$

$p \equiv q$ eq $[(\neg p \vee q) \wedge (\neg q \vee p)]$ eq $(\neg(\neg p \vee q) \vee \neg(\neg q \vee p))$

$p + q$ eq $\neg(\neg p \vee q) \vee \neg(\neg q \vee p)$

p / q eq $\neg p \vee \neg q$

$p \downarrow q$ eq $\neg(p \vee q)$

$M_3 = \{\neg, \supset\}$ este funcțional completă

$p \wedge q$ eq $\neg(p \supset \neg q)$

$p \vee q$ eq $\neg p \supset q$

$p \equiv q$ eq $[(p \supset q) \wedge (q \supset p)]$ eq $\neg[(p \supset q) \supset \neg(q \supset p)]$

$p + q$ eq $[(p \supset q) \supset \neg(q \supset p)]$

p / q eq $p \supset \neg q$

$p \downarrow q$ eq $\neg(\neg p \supset q)$

$M_4 = \{/\}$ este funcțional completă

$\neg p$ eq p/q

$p \wedge q$ eq $\neg(p / q)$ eq $(p / q)/(p / q)$

$p \vee q$ eq $(p / p)/(q / q)$

$p \supset q$ eq $\neg p \vee q$ eq $p/(q / q)$

$p \equiv q$ eq $(p / q)/[(p / p)/(q / q)]$

$p \equiv q$ eq $(p \wedge q) \vee (\neg p \wedge \neg q)$ eq $\neg[(\neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q))]$ eq $\neg[(p / q) \wedge (\neg p / \neg q)]$ eq $(p / q)/(\neg p / \neg q)$ eq $(p / q)/[(p / p)/(q / q)]$.

$p + q$ eq $[p/(q / q)]/[q/(p / p)]$

$p + q$ eq $(p \wedge \neg q) \vee (\neg p \wedge q)$ eq $\neg[\neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)]$ eq $\neg(p \wedge \neg q)/\neg(\neg p \wedge q)$ eq $(p / \neg q)/(\neg p / q)$ eq $[p/(q / q)]/[(p / p)/q]$

$p \downarrow q$ eq $[(p / p)/(q / q)]/[(p / p)/(q / q)]$

$p \downarrow q$ eq $\neg(p \vee q)$ eq $\neg p \wedge \neg q$ eq $\neg\neg(\neg p \wedge \neg q)$ eq $\neg(\neg p / \neg q)$ eq $\neg[(p / p)/(q / q)]$ eq $[(p / p)/(q / q)]/[(p / p)/(q / q)]$

$M_5 = \{\downarrow\}$ este funcțional completă

$\neg p$ eq $p \downarrow p$

$p \wedge q$ eq $(p \downarrow p) \downarrow (q \downarrow q)$

$p \vee q$ eq $(p \downarrow q) \downarrow (p \downarrow q)$

$p \supset q$ eq $[(p \downarrow p) \downarrow q] \downarrow [(p \downarrow p) \downarrow q]$

$p \equiv q$ eq $[(p \downarrow p) \downarrow q] \downarrow [(q \downarrow q) \downarrow p]$

$p \equiv q$ eq $(p \supset q) \wedge (q \supset p)$ eq $(\neg p \vee q) \wedge (\neg q \vee p)$ eq $\neg(\neg p \downarrow q) \wedge \neg(\neg q \downarrow p)$ eq $\neg[(\neg p \downarrow q) \vee \neg(\neg q \downarrow p)]$ eq $(\neg p \downarrow q) \downarrow (\neg q \downarrow p)$ eq $[(p \downarrow p) \downarrow q] \downarrow [(q \downarrow q) \downarrow p]$

$p + q$ eq $[(p \downarrow p) \downarrow (q \downarrow q)] \downarrow (p \downarrow q)$

$p + q$ eq $\neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q)$ eq $\neg\neg[\neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q)]$ eq $\neg\neg[(\neg p \vee \neg q) \wedge (p \vee q)]$ eq $\neg[\neg(\neg p \vee \neg q) \vee \neg(p \vee q)]$ eq $\neg[(\neg p \downarrow \neg q) \vee (p \downarrow q)]$ eq $(\neg p \downarrow \neg q) \downarrow (p \downarrow q)$ eq $[(p \downarrow p) \downarrow (q \downarrow q)] \downarrow (p \downarrow q)$

p / q eq $\neg(p \wedge q)$ eq $\neg p \vee \neg q$ eq $\neg\neg(\neg p \vee \neg q)$ eq $\neg(\neg p \downarrow \neg q)$ eq $(\neg p \downarrow \neg q) \downarrow (\neg p \downarrow \neg q)$ eq $[(p \downarrow p) \downarrow (q \downarrow q)] \downarrow [(p \downarrow p) \downarrow (q \downarrow q)]$.

După cum am arătat mai sus, operatorii / şi $\downarrow$ pot reda, fiecare, orice alt operator al $\mathcal{L}_p$. Mai mult, demonstrăm mai jos următorul rezultat.

Teoremă. Operatorii / şi $\downarrow$ sunt singurii operatori binari care pot constitui, fiecare, mulţimi funcţional complete de operatori fundamentali.

Demonstraţie. Presupunem că o este un operator binar, altul decât / şi $\downarrow$, prin intermediul căruia orice alt operator poate fi definit. Fie *p* şi *q* două variabile propoziţionale. Dacă pentru $p=1$ şi $q=1$ vom avea $p\,o q=1$, atunci orice formulă construită din *p, q* şi operatorul o ar lua întotdeauna valoarea logică 1 dacă variabilele propoziţionale *p* şi *q* ar lua valoarea logică 1. În acest caz negaţia n-ar putea fi definită cu ajutorul acestui operator. Similar, dacă pentru $p=0$ şi $q=0$ vom avea $p\,o q=0$, atunci orice formulă construită numai din variabilele *p* şi *q* şi operatorul o ar fi întotdeauna falsă, dacă ambele variabile iau valoarea logică fals. Nici în acest caz negaţia n-ar putea fi definită utilizând doar operatorul o. De aici rezultă că pentru a putea reda negaţia matricea operatorului o trebuie să aibe următoarea formă:

p	*q*	$p\,o q$
1	1	0
1	0	
0	1	
0	0	1

Dacă în ocurenţele rămase (liniile 2 şi 3) vom pune 1, 1 respectiv 0, 0, atunci vom obţine matricele operatorilor / şi $\downarrow$. În schimb, dacă vom pune celelalte posibilităţi de combinare ale valorilor logice, respectiv 0, 1 sau 1, 0, vom obţine, corespunzător, $\neg p$ şi $\neg q$, adică $p\,o q$ eq $\neg p$ şi $p\,o q$ eq $\neg q$. În aceste din urmă cazuri ar însemna că operatorul binar o poate fi definit doar cu ajutorul negaţiei. Am văzut însă mai sus că negaţia nu poate forma, singură, o mulţime funcţional completă de operatori fundamentali (conjuncţia n-a putut fi definită doar cu ajutorul negaţiei!).

2.1.3.4. Demonstraţii de completitudine funcţională

În paragraful precedent am văzut că anumite mulţimi de operatori pot „traduce" echiveridic orice alt operator. Ele au fost numite mulţimi funcţional complete. De data aceasta vom demonstra câteva rezultate cu privire la completitudinea funcţională.

Teorema 1. $M=\{\neg,\wedge\}$ *este funcţional completă. (Echivalent: orice funcţie de adevăr a* $\mathcal{L}_p$ *poate fi generată de o expresie care conţine strict operatorii negaţie şi conjuncţie).*

Demonstraţie. Fie $\alpha(p_1,...,p_n)$ o funcţie *n*-ară de adevăr. După cum ştim, matricea unei asemenea funcţii va conţine 2^n linii, corespunzătoare celor 2^n interpretări diferite ale variabilelor sale. Numerotăm aceste interpretări de la 1 la 2^n. Fie acum o funcţie f_{ik} definită astfel: $f_{ik} = p_k$, dacă $p_k=1$

$f_{ik} = \neg p_k$, dacă $p_k=0$,

unde *i* enumeră interpretările ($i = 1,...,2^n$), iar *k* enumeră variabilele funcției ($k = 1,...,n$). Din construcția acestei funcții rezultă că, în interpretarea *i*, f_{ik} este adevărată.

Construim acum o funcție g_i, corespunzătoare unei interpretări *i*, astfel:

$$g_i = f_{i1} \wedge ... \wedge f_{in}$$

Din definiția funcției f_{ik} şi din definiția semantică a conjuncției rezultă că funcția g_i este adevărată în interpretarea *i*. Însă g_i este totodată falsă în orice altă interpretare, diferită de interpretarea aleasă, *i*. Acest lucru rezultă din faptul că într-o interpretare *j*, diferită de *i*, cel puțin o funcție f_{ik} are un argument care diferă de valoarea de adevăr pe care p_k o ia în interpretarea *i*; adică funcția f_{ik} ia valoarea logică fals şi cu aceasta întreaga conjuncție $f_{i1} \wedge ... \wedge f_{in}$ devine falsă.

După cum funcția $\alpha(p_1,..., p_n)$ este sau nu o formulă validă a $\mathcal{L}_p$ vom avea, corespunzător, următoarele două cazuri:

1. $\alpha(p_1,..., p_n)$ ia valoarea logică adevărat pentru toate cele 2^n interpretări, adică este o formulă validă. În acest caz $\alpha(p_1,..., p_n)$ poate fi scrisă, echivalent / echiveridic în forma: $\neg(p_1 \wedge ... \wedge \neg p_n) \wedge \neg(p_2 \wedge \neg p_2) \wedge ... \wedge \neg(p_n \wedge \neg p_n)$. Şi astfel, funcția $\alpha(p_1,..., p_n)$ a fost definită utilizând doar $\neg$ şi $\wedge$.

2. $\alpha(p_1,..., p_n)$ nu este o formulă validă. În acest caz fie $i_1,...,i_m$ cele *m* interpretări în care $\alpha(p_1,..., p_n)$ ia valoarea logică *fals*. Construim acum o funcție de adevăr *h* astfel:

$$h = \neg g_{i_1} \wedge ... \wedge \neg g_{i_m}$$

Funcția *h* va fi falsă doar dacă cel puțin un conjunct este fals, echivalent, doar dacă cel puțin o funcție g_{i_r} ($r = 1,...,m$) este adevărată. Echivalent, funcția *h* va fi falsă într-o interpretare i_r în care şi $\alpha(p_1,..., p_n)$ este falsă, motiv pentru care vom pune $\alpha(p_1,..., p_n) = h$. Şi deci şi în acest caz funcția $\alpha(p_1,..., p_n)$ a fost definită doar pe baza $\neg$ şi $\wedge$.

Să revenim acum cu explicații şi exemple la teorema de mai sus.

Să presupunem că funcția de adevăr (i.e. formula) considerată este $\alpha(p,q) = (p_1 \equiv p_2)$ a cărei matrice este:

	p_1	p_2	$p_1 \equiv p_2$
i_1	1	1	1
i_2	1	0	0
i_3	0	1	0
i_4	0	0	1

$i = 1,...,4$

$k = 1,2$; $f_{ik} = p_k$, dacă $p_k = 1$

$f_{ik} = \neg p_k$, dacă $p_k = 0$

Avem, aşadar, pentru linia întâi: $f_{11} = p_1$, $f_{12} = p_2$; pentru linia a doua: $f_{21} = p_1$, $f_{22} = \neg p_2$, pentru linia a treia: $f_{31} = \neg p_1$, $f_{32} = p_2$; pentru linia a patra: $f_{41} = \neg p_1$, $f_{42} = \neg p_2$. Iar acum funcția $g_i = f_{i1} \wedge ... \wedge f_{in}$:

$i = 1$: $g_1 = f_{11} \wedge f_{12} = p_1 \wedge p_2 = 1 \wedge 1 = 1$

$i=2:\ g_2=f_{21}\wedge f_{22}=p_1\wedge\neg p_2=1\wedge 1=1$

$i=3:\ g_3=f_{31}\wedge f_{32}=\neg p_1\wedge p_2=1\wedge 1=1$

.

$i=4:\ g_4=f_{41}\wedge f_{42}=\neg p_1\wedge\neg p_2=1\wedge 1=1$

Aşadar, funcția g_i este adevărată în interpretarea *i*. Să arătăm că g_i este falsă în *orice altă* interpretare. Să luăm funcția g_2 citită în interpretarea 1. În acest caz, $g_2=p_1\wedge\neg p_2=1\wedge 0=0$ (deoarece în interpretarea 1 $p_2=1$ şi deci $\neg p_2=0$). După cum rezultă din acest exemplu, funcția g_i este *falsă* în orice altă interpretare, deoarece cel puțin un conjunct al funcției g_i este fals (deoarece există cel puțin un *k* astfel că p_k ia valoarea logică *opusă* valorii ei din interpretarea *i*) şi astfel întreaga conjuncție este falsă.

Pentru cele două cazuri mai sus menționate:

1. $\alpha(p_1,...,p_n)$ este validă. Fie $\alpha(p_1,p_2)=(p_1\supset p_2)\supset(\neg p_2\supset\neg p_1)$ eq $\neg(\neg p_1\vee p_2)\vee(p_2\vee\neg p_1)$ eq $(p_1\wedge\neg p_2)\vee(p_2\vee\neg p_1)$ eq $(p_1\vee p_2\vee\neg p_1)\wedge(\neg p_2\vee p_2\vee\neg p_1)$ eq $(p_1\vee\neg p_1)\wedge(p_2\vee\neg p_2)$ eq $\neg(p_1\wedge\neg p_1)\wedge\neg(p_2\wedge\neg p_2)$. Şi deci $\alpha(p_1,p_2)$ a fost definită doar prin intermediul $\neg$ şi $\wedge$.

2. $\alpha(p_1,...,p_n)$ nu este validă. Fie $\alpha(p_1,p_2,p_3):\neg[(p_1\equiv p_2)\wedge p_3]$.

	p_1	p_2	p_3	$p_1\equiv p_2$	$(p_1\equiv p_2)\wedge p_3$	$\neg[(p_1\equiv p_2)\wedge p_3]$
i_1	1	1	1	1	1	0
	1	1	0	1	0	1
	1	0	1	0	0	1
	1	0	0	0	0	1
	0	1	1	0	0	1
	0	1	0	0	0	1
i_2	0	0	1	1	1	0
	0	0	0	1	0	1

Cele două interpretări în care α este falsă sunt prima şi a şaptea; le renumerotăm i_1 şi i_2. Construim acum funcția $h=\neg g_{i_1}\wedge...\wedge\neg g_{i_m}$, adică

$h=\neg g_{i_1}\wedge\neg g_{i_2}=\neg(p_1\wedge p_2\wedge p_3)\wedge\neg(\neg p_1\wedge\neg p_2\wedge p_3)=\alpha(p_1,p_2,p_3)$. Să arătăm acum, prin transformări sintactice, că $\alpha(p_1,p_2,p_3)$ poate fi redată în expresia de mai sus, care conține doar operatorii $\neg$ şi $\wedge$.

$\alpha(p_1,p_2,p_3)=\neg[(p_1\equiv p_2)\wedge p_3]$ eq $\neg\{[(p_1\wedge p_2)\vee(\neg p_1\wedge\neg p_2)]\wedge p_3\}$ eq $\neg[(p_1\wedge p_2\wedge p_3)\vee(\neg p_1\wedge\neg p_2\wedge p_3)]$ eq $\neg(p_1\wedge p_2\wedge p_3)\wedge\neg(\neg p_1\wedge\neg p_2\wedge p_3)$.

În cazul funcției $\alpha(p_1,p_2)=(p\equiv q)$ procedăm similar.

Adică $h=\neg g_{i_1}\wedge\neg g_{i_2}=\neg(p_1\wedge\neg p_2)\wedge\neg(\neg p_1\wedge p_2)$eq $(\neg p_1\vee p_2)\wedge(p_1\vee\neg p_2)$ eq $(p_1\supset p_2)\wedge(p_2\supset p_1)$ eq $p_1\equiv p_2$.

Teorema 1 exprimă un rezultat fundamental cu privire la completitudinea funcțională a mulțimilor de operatori. Fiindcă conjuncția poate fi definită utilizând $\neg$ şi $\vee$ sau $\neg$ şi $\supset$ (i.e. $(p\wedge q)=_{df}\neg(\neg p\vee\neg q)=_{df}\neg(p\supset\neg q)$, rezultă că $M=\{\neg,\vee\}$ şi $M=\{\neg,\supset\}$ formează, de asemenea, mulțimi funcțional complete de operatori. Mai mult, definițiile $\neg p=_{df}p/p$ şi

$p \wedge q =_{df} \neg(p/q)$ ne arată că $M = \{/\}$ formează o astfel de mulțime funcțional completă. Iar definițiile $\neg p =_{df} p \downarrow p$ şi $p \vee q =_{df} \neg(p \downarrow q)$ ne arată că şi $\neg$ şi $\vee$ pot fi reduse la $\downarrow$. Şi fiindcă $M = \{\neg, \vee\}$ este o mulțime funcțional completă de operatori, rezultă că şi $M = \{\downarrow\}$ formează o astfel de mulțime.

Cum $M = \{\neg, \wedge\}$ şi $M = \{\neg, \vee\}$ sunt mulțimi funcțional complete, conchidem că $M = \{\neg, \wedge, \vee\}$ este funcțional completă. Deşi acest lucru rezultă cu claritate din cele de mai sus, vom demonstra acest fapt într-o teoremă al cărei rezultat este paralel celui conținut în *Teorema 1*.

Teorema 2. $M = \{\neg, \wedge, \vee\}$ *este funcțional completă. (Echivalent: orice funcție de adevăr a* $\mathcal{L}_p$ *poate fi generată de o expresie care conține strict operatorii negație, conjuncție şi disjuncție).*

Demonstrație. Vom considera din nou funcțiile f_{ik} şi g_i, aşa cum au fost definite în Teorema 1. După cum funcția $\alpha(p_1,..., p_n)$ este sau nu o formulă *nesatisfiabilă* vom avea, din nou, două cazuri:

1. $\alpha(p_1,..., p_n)$ este o formulă nesatisfiabilă. În acest caz $\alpha(p_1,..., p_n)$ poate fi scrisă echivalent / echiveridic în forma: $(p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2) \vee ... \vee (p_n \wedge \neg p_n)$.

Exemplu. $\alpha(p_1,..., p_n)$: $\neg\{[(p_1 \supset p_2) \wedge (p_2 \supset p_3)] \supset (p_1 \supset p_3)\}$

$\neg\{[(p_1 \supset p_2) \wedge (p_2 \supset p_3)] \supset (p_1 \supset p_3)\}$ eq $\neg\{\neg[(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3)] \vee (\neg p_1 \vee p_3)\}$ eq $[(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3)] \wedge \neg(\neg p_1 \vee p_3)$ eq $[(\neg p_1 \vee p_2) \wedge (\neg p_2 \vee p_3)] \wedge p_1 \wedge \neg p_3$ eq $(\neg p_1 \wedge \neg p_2 \wedge p_1 \wedge \neg p_3) \vee (\neg p_1 \wedge p_3 \wedge p_1 \wedge \neg p_3) \vee (p_2 \wedge \neg p_2 \wedge p_1 \wedge \neg p_3) \vee (p_2 \wedge p_3 \wedge p_1 \wedge \neg p_3)$ eq $(p_1 \wedge \neg p_1) \vee (p_3 \wedge \neg p_3) \vee (p_2 \wedge \neg p_2) \vee (p_3 \wedge \neg p_3)$ eq $(p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2) \vee (p_3 \wedge \neg p_3)$.

2. $\alpha(p_1,..., p_n)$ este o formulă satisfiabilă (în sens larg). În acest caz, din cele 2^n interpretări vom alege, de data aceasta, acele interpretări (i.e. linii) în care $\alpha(p_1,..., p_n)$ ia valoarea logică adevărat. Fie $i_1,...,i_l$ toate aceste interpretări. Construim acum o funcție

$$h^* = g_{i_1} \vee ... \vee g_{i_l}$$

şi punem, simplu, $\alpha(p_1,..., p_n) = h^*$.

Exemplu. $\alpha(p_1,..., p_n)$: $\neg[(p_1 \equiv p_2) \wedge p_3]$, adică formula luată la exemplul Th1, cazul 2. În acest caz $\alpha(p_1,..., p_n) = h^* =$

$(p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee$
$\vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)$. Putem verifica acest lucru, obținând acelaşi rezultat prin transformări sintactice corecte ale formule $\alpha(p_1,..., p_n)$.

$\neg[(p_1 \equiv p_2) \wedge p_3]$ eq $\neg\{[(p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2) \wedge p_3]\}$ eq $[\neg(p_1 \wedge p_2) \wedge \neg(\neg p_1 \wedge \neg p_2)] \vee \neg p_3$ eq $[(\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2)] \vee \neg p_3$ eq $(\neg p_1 \wedge p_1) \vee (\neg p_1 \wedge p_2) \vee (\neg p_2 \wedge p_1) \vee (\neg p_2 \wedge p_2) \vee \neg p_3$ eq $(\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2) \vee \neg p_3$.

Pentru a ajunge la rezultatul căutat va trebui ca în fiecare disjunct să introducem variabilele care lipsesc. În exemplul nostru, în primi doi disjuncți lipseşte variabila p_3 iar în ultimul lipsesc variabilele p_1 şi p_2. Dacă *D* este disjunctul considerat, atunci introducerea unei variabile p_i se face după următoarea relație:

D eq $D \wedge (p_i \vee \neg p_i)$ eq $(D \wedge p_i) \vee (D \wedge \neg p_i)$.

Faptul că această relație are loc este uşor de remarcat. Căci $p_i \vee \neg p_i$ ia întotdeauna valoarea logică adevărat, iar adăugarea unui argument adevărat unei conjuncții (D fiind o *conjuncție* de variabile propoziționale negate sau nenegate) nu-i alterează valoarea de adevăr.

Pe baza acestei relații avem:

$\neg p_1 \wedge p_2$ eq $(\neg p_1 \wedge p_2) \wedge (p_3 \vee \neg p_3)$ eq $(\neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3)$

$p_1 \wedge \neg p_2$ eq $(p_1 \wedge \neg p_2 \wedge p_3) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3)$

$\neg p_3$ eq $(\neg p_3 \wedge p_1) \vee (\neg p_3 \vee \neg p_1)$ eq

eq $(\neg p_3 \wedge p_1 \wedge p_2) \vee (\neg p_3 \wedge p_1 \wedge \neg p_2) \vee (\neg p_3 \wedge \neg p_1 \wedge p_2) \vee (\neg p_3 \wedge \neg p_1 \wedge \neg p_2)$

În fine $\alpha(p_1,..., p_n) = (\neg p_1 \wedge p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3) \vee$

$\vee (p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3)$, de unde, prin reordonarea disjuncților (pe baza comutativității), după ce am eliminat disjuncții care se repetă, obținem rezultatul dorit.

Exerciții.

1. Se dă următorul şir de valori de adevăr, care reprezintă valorile unei funcții de adevăr binare $\alpha(p_1, p_2)$, în asignarea standard: 1100.

a) Să se determine, pe baza *Th* 1 şi *Th* 2 de mai sus, două expresii, una care conține strict operatorii $\neg$ şi $\wedge$ iar cealaltă operatorii $\neg$, $\wedge$ şi $\vee$ şi care pot genera funcția dată.

b) Să se demonstreze, prin transformări sintactice corecte, echiveridicitatea $\alpha(p_1, p_2)$ eq p_1, în ambele cazuri.

Soluție. $\alpha(p_1, p_2)$ este prependența, adică

p_1	p_2	$\alpha(p_1, p_2)$
1	1	1
1	0	1
0	1	0
0	0	0

a) În acord cu *Th* 1 $\alpha(p_1, p_2) = h = \neg(\neg p_1 \wedge p_2) \wedge \neg(\neg p_1 \wedge \neg p_2)$, iar în acord cu Th 2, $\alpha(p_1, p_2) = h^* = (p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2)$.

b) $\neg(\neg p_1 \wedge p_2) \wedge \neg(\neg p_1 \wedge \neg p_2)$ eq $(p_1 \vee \neg p_2) \wedge (p_1 \vee p_2)$ eq $p_1 \vee (p_2 \wedge \neg p_2)$ eq p_1.

Similar pentru cealaltă expresie.

2. Aceleaşi cerințe ca la exercițiu 1 pentru funcțiile $\neg p$, q şi $\neg q$.

3. Se dau următoarele funcții de adevăr:

α_1[9] $= [(p/q) \supset \neg r] \downarrow (p \wedge r)$

$\alpha_2 = [(p+q) \supset r] / [(p \downarrow \neg q) \vee r]$

$\alpha_3 = [(p/q) \supset r] \vee (p \wedge r)$

$\alpha_4 = [(p \supset q) \wedge (p \supset r)] \supset (p \vee r)$

[9] În cele ce urmează vom omite menționarea argumentelor după simbolul „α”, iar pentru variabile vom utiliza simbolurile p, q,..., în loc de p_1, p_2,...

$\alpha_5 = (p \wedge q) \supset [(p + q) \wedge (\neg p \vee q)]$

a) Construiți matricele acestor funcții de adevăr.

b) Determinați, în fiecare caz, două expresii, una care conține strict operatorii $\neg$ şi $\wedge$ iar cealaltă operatorii $\neg$, $\wedge$ şi $\vee$ şi care pot genera funcția dată.

c) Verificați acest fapt prin transformări sintactice adecvate.

d) Pentru α_1 construiți o expresie echiveridică în care operatorul „$\downarrow$” nu mai apare; pentru α_2, una în care „$+$” şi „$\supset$” nu mai apar; pentru α_3, o expresie care nu conține „$\vee$” iar pentru α_4 şi α_5 construiți expresiile echiveridice corespunzătoare care nu conțin „$\supset$”.

2.1.4. Procedee de decizie în $\mathcal{L}_p$

Aşa cum am văzut în paragraful 2.1.2.1, formulele $\mathcal{L}_p$ pot fi formule întotdeauna adevărate (valide), uneori adevărate (satisfiabile) sau niciodată adevărate (nesatisfiabile). În practică este necesar ca, de fiecare dată, să putem spune cărei clase îi aparține o formulă oarecare a $\mathcal{L}_p$. Înainte de toate ne interesează să putem decide dacă o formulă a $\mathcal{L}_p$ este validă sau nu.

Definiție 1. A decide în $\mathcal{L}_p$ înseamnă a găsi un procedeu efectiv prin care, într-un număr finit de paşi, să se poată stabili dacă o formulă arbitrară a $\mathcal{L}_p$ este validă sau nu.

Aşa cum vom vedea mai jos, prin aplicarea oricăruia din procedeele descrise vom putea stabili, în cazul în care formula este nevalidă, dacă formula respectivă este satisfiabilă sau nu.

2.1.4.1. Procedeul matriceal

Procedeul matriceal este cel descris în paragraful 2.1.2.3. Decizia cu ajutorul acestui procedeu o facem în acord cu definițiile 1 – 3 ale acelui paragraf. Să mai considerăm un exemplu simplu.

Fie $\alpha : [(p / \neg q) \supset r] \supset [(p \downarrow \neg r) \wedge q]$

În acord cu algoritmul prezentat în 2.1.2.3 matricea formulei α este:

p	q	r	$\neg q$	$p / \neg q$	$(p / \neg q) \supset r$	$\neg r$	$p \downarrow \neg r$	$(p \downarrow \neg r) \wedge q$	α
1	1	1	0	1	1	0	0	0	0
1	1	0	0	1	0	1	0	0	1
1	0	1	1	0	1	0	0	0	0
1	0	0	1	0	1	1	0	0	0
0	1	1	0	1	1	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	0	1	1	1	1	0	1	0	0
0	0	0	1	1	0	1	0	0	1

În acord cu definiția 3 formula α este satisfiabilă (realizabilă), deoarece coloana finală, care redă valorile logice ale formulei α, pentru toate cele 8 interpretări ale variabilelor ei, conține valori *neomogene* de adevăr.

2.1.4.2. Reductio test

Indiscutabil, procedeul matriceal devine inconvenabil atunci când numărul variabilelor propoziționale creşte. De aceea, pentru verificarea validității unei formule a $\mathcal{L}_p$, putem proceda prin reducere la absurd, în următorul fel. Presupunem că formula dată, α, este nevalidă. Există, aşadar, o interpretare a variabilelor ei care falsifică formula α. Dacă în încercarea de a găsi o astfel de interpretare ajungem la o contradicție, atunci formula nu poate fi falsificată şi deci este o formulă validă. În caz contrar, α este nevalidă. Acest procedeu se numeşte *reductio test*.

Exemplu.

$\alpha : (p \supset q) \supset [(r \vee p) \supset (r \vee q)]$

Presupunem că α poate fi falsificată. Aceasta înseamnă că $(p \supset q) = 1$, $(r \vee p) = 1$ şi $(r \vee q) = 0$. Din falsitatea formulei $r \vee q$ deducem $r = 0$ şi $\mathbf{q = 0}$. Cum $(r \vee p) = 1$ iar $r = 0$, rezultă că $\mathbf{p = 1}$. Dacă transferăm aceste valori logice în antecedentul implicației, $p \supset q$, rezultă că implicația $p \supset q$ este adevărată (în acord cu presupunerea făcută) atunci când $p = 1$ şi $q = 0$. Contradicție! Şi deci α nu poate fi falsificată, echivalent α este o formulă validă a $\mathcal{L}_p$.

Exerciții. Decideți cu ajutorul *reductio test* asupra următoarelor formule:

$\alpha_1 : [(p \supset q) \wedge (q \supset r)] \supset (p \supset r)$

$\alpha_2 : [(p / q) \wedge r] \vee (p \wedge \neg r)$

$\alpha_3 : (p + q) \wedge \neg q \wedge r$

$\alpha_4 : [(p \wedge q) \supset r] \supset (p \supset r)$

2.1.4.3. Procedeul Quine

Este un procedeu de decizie mai elegant decât cel matriceal, mai ales atunci când numărul variabilelor propoziționale este mare.

Procedeul Quine[10] se bazează pe *asignarea succesivă* de valori de adevăr variabilelor formulei considerate şi pe aplicarea regulilor de adevăr ale operatorilor logici din formulă. De regulă, pentru a simplifica cât mai mult formula care trebuie evaluată se alege mai întâi variabila care apare cel mai frecvent şi i se atribuie, alternativ, valorile logice 1, respectiv 0. Dacă în felul acesta nu se ajunge la o valoare de adevăr determinată, procedeul continuă cu variabila următoare, până când se obțin *doar* valori de adevăr. Decizia o vom face similar procedeului matriceal, adică pe baza definițiilor 1 – 3 din 2.1.2.3.

Formularea regulilor de adevăr ale operatorilor logici o facem pe baza matricei operatorului respectiv. Fiecare operator este explicitat prin două reguli, una pentru cazul în care un argument este adevărat iar cealaltă pentru cazul în care un argument este fals. Prin aplicarea unei reguli asupra formulei care-l conține, formula se reduce fie la o valoare de adevăr, fie la celălalt argument negat sau nenegat.

[10] Cf. W.v.O. Quine, *Methods of Logic*, Holt, Rinehort and Winston, Inc., 1972.

Regulile de adevăr ale operatorilor logici.

$\wedge$: $(\alpha \wedge \beta)$. R_1. Dacă într-o conjuncție un argument este *adevărat*, atunci valoarea logică a conjuncției se reduce la valoarea celuilalt argument.

R_2. Dacă într-o conjuncție un argument este *fals*, atunci valoarea logică a conjuncției se reduce la fals.

$\vee$: $(\alpha \vee \beta)$ (Exercițiu)

$\supset$: $(\alpha \supset \beta)$ R_1. Dacă într-o implicație un argument este *adevărat*, atunci valoarea logică a implicației se reduce la valoarea consecventului.

R_2. Dacă într-o implicație un argument este *fals*, atunci valoarea logică a implicației se reduce la valoarea *negată* a antecedentului.

Formularea acestor reguli pentru implicație o facem cu ușurință comparând, în matricea implicației, valorile funcției cu valorile argumentelor ei. Pentru R_1, de exemplu, luăm cazurile (i.e. liniile) în care (cel puțin) un argument este adevărat, adică primele trei linii. Observăm că valorile funcției, pentru primele trei categorii de asignări, coincid cu valorile consecventului. Pentru R_2 procedăm similar: luăm liniile în care (cel puțin) un argument este fals, adică liniile 2, 3 şi 4, şi observăm că valorile funcției coincid cu valorile *negate* ale antecedentului. La fel procedăm şi în stabilirea regulilor operatorilor $\subset, \not\subset$ şi # (Exercițiu).

$\equiv$: $(\alpha \equiv \beta)$ R_1. Dacă într-o echivalență un argument este *adevărat*, atunci valoarea logică a echivalenței se reduce la valoarea celuilalt argument.

R_2. Dacă într-o echivalență un argument este *fals*, atunci valoarea logică a echivalenței se reduce la valoarea *negată* a celuilalt argument.

$+$: $(\alpha + \beta)$ (Exercițiu)

$/$: (α / β) R_1. Dacă într-o incompatibilitate un argument este adevărat, atunci valoarea logică a incompatibilității se reduce la valoarea negată a celuilalt argument.

R_2. Dacă într-o incompatibilitate un argument este fals, atunci valoarea logică a incompatibilității se reduce la adevărat (de ce?)

$\downarrow$: $(\alpha \downarrow \beta)$ (Exercițiu)

Remarcă.

Cele şase funcții din tabelul lui Wittgenstein $(p, q, \neg p, \neg q, \mathcal{T}, \mathcal{C})$ rămân în afara considerațiilor, deoarece ele nu corelează în vreun fel valorile logice ale argumentelor lor.

Exemplu. Fie $\alpha : [(p \wedge q) \supset r] \supset (p / \neg q)$.

$p = 1$

$[(1 \wedge q) \supset r] \supset (1 / \neg q)$

$(q \supset r) \supset q$

$q = 1$ | $q = 0$

$(1 \supset r) \supset 1$ | $(0 \supset r) \supset 0$

$r \supset 1$ | $1 \supset 0$

1 | 0

$p = 0$

$[(0 \wedge q) \supset r] \supset (0 / \neg q)$

$(0 \supset r) \supset 1$

$1 \supset 1$

1

Exerciții. Decideți cu ajutorul procedeului Quine asupra următoarelor formule ale $\mathcal{L}_p$:

$\alpha_1 : (p \supset q) \supset [(q \supset r) \supset (p \supset r)]$
$\alpha_2 : [(p \supset q) \supset (q \supset r)] \supset (p \supset r)$
$\alpha_3 : [(p / q) \supset \neg q] + [(\neg p / s) \vee t]$
$\alpha_4 : [\,(p \downarrow q) \vee r\,] \wedge [(t \equiv p) \wedge \neg q]$
$\alpha_5 : [(p \supset q) \wedge (p \supset \neg q)] \supset \neg p$
$\alpha_6 : [(p / q) \supset (r / \neg r)] \supset (s \wedge \neg s)$
$\alpha_7 : [(p \equiv q) \wedge (r \equiv s)] \supset [(p \wedge r) \equiv (q \wedge s)]$

Remarcă. Arătați că formula din α_7 rămâne o formulă validă a $\mathcal{L}_p$ dacă operatorul „$\wedge$” din consecventul implicației este înlocuit cu oricare din următorii operatori binari ai $\mathcal{L}_p$: $\vee, \supset, \subset, \equiv, +, /, \downarrow, \not\subset, \#$.

2.1.4.4. Formele normale în $\mathcal{L}_p$

Definiție 1. Formele normale sunt expresii a căror structură sintactică ne permite să decidem asupra unei formule a $\mathcal{L}_p$.

Aşadar, atunci când decidem cu ajutorul formelor normale nu mai asignăm valori de adevăr variabilelor propoziționale, ci construim nişte expresii, logic echivalente formulei asupra căreia vrem să decidem, şi decidem doar pe baza configurațiilor de simboluri. În toate transformările sintactice efectuate dimensiunea semantică e una implicită.

În logică, în general, formele normale sunt de o mare varietate. Lucrarea de față tratează doar cazul în care în aceste construcții apar doar operatorii $\neg, \wedge, \vee$, numiți operatori booleeni (motiv pentru care ele se mai numesc şi forme normale booleene).

Formele normale booleene se constituie în două clase distincte, după cum operatorul principal (cel care dă numele expresiei) este conjuncția sau disjuncția.

Definiție 2. O formulă α a $\mathcal{L}_p$ este în forma normală conjunctivă (abreviat: α_c) dacă are forma unei conjuncții $\alpha_1 \wedge \ldots \wedge \alpha_n \ (n \geq 1)$, în care fiecare conjunct $\alpha_i \ (i = 1, \ldots, n)$ are forma unei disjuncții $\beta_{i_1} \vee \ldots \vee \beta_{i_m} \ (m_i \geq 1)$, ai cărei membri sunt variabile propoziționale nenegate sau negate o singură dată.

Teorema 1. Orice formulă a $\mathcal{L}_p$ admite o formă normală conjunctivă logic echivalentă (i. e. $\models \alpha \equiv \alpha_c$).

Demonstrație. Demonstrația acestei teoreme o vom face, simplu, indicând *algoritmul* aducerii unei formule date la forma normală conjunctivă. Algoritmul are următorii paşi:

1. Se înlocuiesc toate subformulele care conțin alți operatori decât operatorii $\neg, \wedge, \vee$ cu expresii echivalente care conțin doar aceşti operatori. În felul acesta obținem o formulă, echivalentă formulei date, dar care conține numai operatori booleeni. De exemplu, dacă α este formula $[(p \supset q) \wedge \neg r] \vee (q / \neg q)$, atunci avem, corespunzător, subformulele $p \supset q$ şi $q / \neg q$ care trebuie transformate echivalent în $\neg p \vee q$ şi $\neg(q \wedge \neg q)$. Şi obținem astfel formula $\alpha_1 : [(\neg p \vee q) \wedge \neg r] \vee \neg(q \wedge \neg q)$, echivalentă formulei α; aşadar $\models \alpha \equiv \alpha_1$.

2. Se înlocuiesc toate subformulele în care negația neagă o expresie care conține un operator binar, cu expresii echivalente în care negația neagă numai variabile propoziționale.

În α_1, $\neg(q \wedge \neg q)$ devine $\neg q \vee \neg\neg q$, şi deci α_1 devine, echivalent, $\alpha_2 : [(\neg p \vee q) \wedge \neg r] \vee (\neg q \vee \neg\neg q)$. Şi deci $\models \alpha_1 \equiv \alpha_2$.

3. Se înlocuiesc toate subformulele de forma $\neg\neg p_i$, unde p_i este o variabilă propozițională, cu p_i. Firește, dacă avem mai multe negații, eliminarea lor se face astfel încât ceea ce păstrăm este o variabilă propozițională nenegată sau negată o singură dată (cf. formulelor 1 şi 2, § 2.1.2.3). Şi obținem astfel $\alpha_3 : [(\neg p \vee q) \wedge \neg r] \vee (\neg q \vee q)$, astfel că $\models \alpha_2 \equiv \alpha_3$.

4. Se înlocuiesc succesiv toate acele subformule de forma $\gamma_1 \vee (\gamma_2 \wedge \gamma_3)$ cu expresiile echivalente corespunzătoare, de forma $(\gamma_1 \vee \gamma_2) \wedge (\gamma_1 \vee \gamma_3)$. Şi în felul acesta din α_3 obținem formula echivalentă ei: $\alpha_c : (\neg p \vee q \vee \neg q \vee q) \wedge (\neg r \vee \neg q \vee q)$, şi care este, în acord cu Def. 2, forma normală conjunctivă a formulei inițiale α. Şi astfel $\models \alpha_3 \equiv \alpha_c$. De unde, pe baza tranzitivității echivalenței obținem $\models \alpha \equiv \alpha_c$.

Remarcă. În aceste transformări echiveridice adesea aplicarea algoritmului nu este obligatorie, deoarece se pot executa simultan mai mulți paşi ai algoritmului.

Teorema următoare ne spune cum stabilim cu ajutorul formelor normale conjunctive dacă o formulă a $\mathcal{L}_p$ este validă sau nu.

Teorema 2. O formulă a $\mathcal{L}_p$, aflată în forma normală conjunctivă, este o formulă validă a $\mathcal{L}_p$ ddacă fiecare conjunct conține cel puțin o variabilă propozițională atât negată cât şi nenegată.

Demonstrație. Dacă formula α este validă, atunci ea este adevărată în orice interpretare a variabilelor sale. Cum $\models \alpha \equiv \alpha_c$ şi $\models \alpha$, rezultă că α_c este o formulă validă, fapt evident deoarece fiecare conjunct conține formula validă $p_i \vee \neg p_i$. Reciproc, dacă fiecare conjunct al formulei α_c are forma $p_1 \vee ... \vee p_i \vee \neg p_i \vee ... \vee p_m$, atunci, pe baza matricei operatorului $\vee$ conchidem că un astfel de conjunct este întotdeauna adevărat (i.e. valid), deoarece conține disjuncția validă $p_i \vee \neg p_i$. Şi astfel, indiferent de asignările făcute celorlalte variabile propoziționale din conjunct, el este unul valid. Şi astfel α_c este o formulă validă. Însă α este echivalentă cu α_c. Şi deci α este o formulă validă.

Dacă, dimpotrivă, α_c nu are forma menționată de teoremă, adică cel puțin un conjunct nu conține o variabilă propozițională atât negată cât şi nenegată, atunci α este nevalidă (argumentați).

Definiție 3. O formulă α a $\mathcal{L}_p$ este în forma normală conjunctivă perfectă (abreviat: α_c^x) dacă este în forma normală conjunctivă şi fiecare conjunct conține toate variabilele propoziționale ale formulei, negate sau nenegate.

Teorema 3. Orice formulă a $\mathcal{L}_p$ admite o formă normală conjunctivă perfectă logic echivalentă (i.e $\models \alpha \equiv \alpha_c^x$).

Demonstrație. Vom proceda ca mai sus, la Th. 1, indicând algoritmul aducerii unei formule date la forma normală conjunctivă perfectă. Potrivit teoremei 1, formula α admite forma normală conjunctivă α_c. Dacă unul dintre conjuncți, α_i nu conține variabila

propoziţională p_k, atunci această variabilă se introduce în conjunctul α_i pe baza următoarei transformări echivalente: $\alpha_i \equiv [\alpha_i \vee (p_k \wedge \neg p_k)] \equiv [(\alpha_i \vee p_k) \wedge (\alpha_i \vee \neg p_k)]$. Pe baza acestor echivalenţe putem introduce, în fiecare conjunct, toate variabilele formulei α. Forma normală conjunctivă perfectă astfel obţinută este echivalentă cu α_c şi deci cu α (prin tranzitivitatea „≡”). Aşadar, o formulă α a $\mathcal{L}_{\mathrm{p}}$ este validă ddacă forma ei normală conjunctivă perfectă este validă.

În formula α_c, din exemplul de mai sus, în primul conjunct lipseşte *r* iar în al doilea *q*. Introducem aceste variabile, corespunzător, în modul mai sus indicat.

Primul conjunct, $\neg p \vee q \vee \neg q \vee q$, devine, echivalent, $(\neg p \vee q \vee \neg q \vee q) \vee (r \wedge \neg r)$, respectiv $(\neg p \vee q \vee \neg q \vee q \vee r) \wedge (\neg p \vee q \vee \neg q \vee q \vee \neg r)$. Al doilea, $\neg r \vee \neg q \vee q$, devine $(\neg r \vee \neg q \vee q) \vee (p \wedge \neg p)$, adică $(\neg r \vee \neg q \vee q \vee p) \wedge (\neg r \vee \neg q \vee q \vee \neg p)$. Şi astfel,
$\alpha_c^x : (\neg p \vee q \vee \neg q \vee q \vee r) \wedge (\neg p \vee q \vee \neg q \vee q \vee \neg r) \wedge (\neg r \vee \neg q \vee q \vee p) \wedge (\neg r \vee \neg q \vee q \vee \neg p)$.

Definiţie 4. O formulă α a $\mathcal{L}_{\mathrm{p}}$ este în forma normală disjunctivă (abreviat: α_d) dacă are forma unei disjuncţii $\alpha_1 \vee \ldots \vee \alpha_n$ $(n \geq 1)$, în care fiecare disjunct α_i $(i = 1,\ldots,n)$ are forma unei conjuncţii $\beta_{i_1} \wedge \ldots \wedge \beta_{i_m}$ $(m_i \geq 1)$, ai cărei membri sunt variabile propoziţionale nenegate sau negate o singură dată.

Teorema 4. Orice formulă a $\mathcal{L}_{\mathrm{p}}$ admite o formă normală disjunctivă logic echivalentă (i.e. $\models \alpha \equiv \alpha_d$).

Demonstraţie. Similar teoremei 1, vom indica algoritmul aducerii unei formule date la forma normală disjunctivă. Primii 3 paşi sunt cei menţionaţi la teorema 1.

Pasul 4. Se înlocuiesc succesiv toate acele subformule de forma $\gamma_1 \vee (\gamma_2 \wedge \gamma_3)$ cu expresiile echivalente corespunzătoare, de forma $(\gamma_1 \wedge \gamma_2) \vee (\gamma_1 \wedge \gamma_3)$. În felul acesta obţinem forma normală disjunctivă, α_d, a formulei α, astfel că $\models \alpha \equiv \alpha_d$.

Din formula din exemplul precedent (din pasul 3) $[(\neg p \vee q) \wedge \neg r] \vee (\neg q \vee q)$ obţinem astfel, în pasul 4, $\alpha_d : (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee \neg q \vee q$.

Dacă prin intermediul formelor normale conjunctive putem decide dacă o formulă este validă sau nu, cu ajutorul formelor normale disjunctive putem testa *nesatisfiabilitatea* formulei în cauză.

Teorema 5. O formulă a $\mathcal{L}_{\mathrm{p}}$, aflată în forma normală disjunctivă, este o formulă nesatisfiabilă a $\mathcal{L}_{\mathrm{p}}$ ddacă fiecare disjunct conţine cel puţin o variabilă propoziţională atât negată cât şi nenegată.

Demonstraţie. Dacă formula α este nesatisfiabilă, atunci α ia valoarea logică fals pentru orice interpretare a variabilelor sale. Cum $\models \alpha \equiv \alpha_d$, rezultă că şi α_d este nesatisfiabilă. Însă cum α_d este o disjuncţie de conjuncţi, rezultă că fiecare disjunct este nesatisfiabil, fapt evident din construcţia lui sintactică, deoarece este o conjuncţie care conţine conjuncţia nesatisfiabilă $p_i \wedge \neg p_i$. Reciproc (similar).

Dacă, dimpotrivă, α_d nu are forma menţionată de teoremă, adică cel puţin un disjunct nu conţine o variabilă propoziţională împreună cu negaţia ei, atunci formula nu este nesatisfiabilă (de ce?).

Exemplu. $\alpha : [\neg(p \supset q) \supset \neg q] \supset [(q / r) \wedge (q \wedge r)]$

Aducem formula α la forma normală disjunctivă.

1. $\{\neg[\neg\neg(\neg p \vee q) \vee \neg q] \vee [\neg(q \wedge r) \wedge (q \wedge r)]\} \equiv \neg(\neg p \vee q \vee \neg q) \vee [\neg(q \wedge r) \wedge (q \wedge r)]$
2. $(\neg\neg p \wedge \neg q \wedge \neg\neg q) \vee [(\neg q \vee \neg r) \wedge (q \wedge r)]$
3. $(p \wedge \neg q \wedge q) \vee [(\neg q \vee \neg r) \wedge (q \wedge r)]$
4. $[(p \wedge \neg q \wedge q) \vee (\neg q \wedge q \wedge r) \vee (\neg r \wedge q \wedge r)] \equiv \alpha_d$

Aşadar, formula α este nesatisfiabilă.

Remarcă 1. Dacă o formulă α a $\mathcal{L}_p$ nu satisface nici teorema 2 şi nici teorema 5, atunci α este o formulă satisfiabilă.

Definiţie 5. O formulă α *a* $\mathcal{L}_p$ *este în forma normală disjunctivă perfectă (abreviat:* α_d^x*) dacă este în forma normală disjunctivă şi fiecare disjunct conţine toate variabilele propoziţionale ale formulei, negate sau nenegate.*

Teorema 6. Orice formulă a $\mathcal{L}_p$ *admite o formă normală disjunctivă perfectă logic echivalentă (i.e*$\models \alpha \equiv \alpha_d^x$*).*

Demonstraţie. Vom proceda similar demonstraţiei teoremei 3 îndicând algoritmul construcţiei formulei α_d^x. Potrivit teoremei 4, orice formulă a $\mathcal{L}_p$ admite o formă normală disjunctivă logic echivalentă. Dacă, o dată construită α_d, unul dintre conjuncţi, α_i, nu conţine o variabilă propoziţională p_j, atunci vom introduce această variabilă în disjunctul respectiv, pe baza următoarelor echivalenţe: $\alpha_i \equiv [\alpha_i \wedge (p_j \vee \neg p_j)] \equiv [(\alpha_i \wedge p_j) \vee (\alpha_i \wedge \neg p_j)]$. În felul acesta putem introduce, în fiecare disjunct, variabilele care apar în formula α, obţinând astfel α_c^x astfel încât $\models \alpha \equiv \alpha_c^x$.

În exemplul de mai sus, fiecare dintre cei trei disjuncţi are o variabilă lipsă. În primul disjunct, de exemplu, putem introduce variabila *r* astfel:

$(p \wedge \neg q \wedge q) \equiv [(p \wedge \neg q \wedge q) \wedge (r \vee \neg r)] \equiv [(p \wedge \neg q \wedge q \wedge r) \vee (p \wedge \neg q \wedge q \wedge \neg r)]$.

Similar procedăm cu ceilalţi disjuncţi şi, în final, construim α_d^x (exerciţiu).

În fine, încheiem acest paragraf cu câteva consideraţii privitoare la *relaţia dintre formele normale perfecte şi relaţia de consecinţă semantică.*

Pe baza formelor normale perfecte putem determina:

a) toate consecinţele semantice ale unei formule date
b) toate premisele unei formule date

În primul caz, construim mai întâi *forma normală conjunctivă perfectă*, α_c^x, a formulei date, α. Consecinţele semantice ale formulei α vor fi:

a) fiecare conjunct al α_c^x
b) fiecare conjuncţie de conjuncţi din α_c^x
c) formula ca atare, α_c^x.

În al doilea caz, construim mai întâi *forma normală disjunctivă perfectă*, α_d^x, a formulei date, α. Vor fi premise ale formulei:

a) fiecare disjunct al α_d^x
b) fiecare disjuncţie de disjuncţi din α_d^x

c) formula ca atare, α_d^x.

Remarcă 2. O dată construită matricea unei formule, formele normale perfecte pot fi, la rândul lor, construite direct de pe matrice. Fie, de exemplu, formula α din 2.1.4.1. Dacă din matricea acestei formule alegem acele linii în care α ia valoarea logică *adevărat*, atunci *forma normală disjunctivă perfectă* a formulei se obține legând conjunctiv variabilele fiecărei linii dintre cele alese (în raport cu valoarea logică a variabilei respective) şi, apoi, conectând disjunctiv liniile. Pentru formula din 2.1.4.1. avem:

$\alpha_d^x : (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$

Pentru construirea *formei normale conjunctive perfecte* procedăm după cum urmează. Alegem acele linii în care α ia valoarea logică *fals*, legăm disjunctiv variabilele propoziționale în *raport invers* cu valoarea lor logică, iar apoi conectăm conjunctiv aceste linii. Pentru formula din 4.1 avem:

$\alpha_c^x : (\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

Dacă aducem formula α la α_d^x şi α_c^x vom obține, corespunzător, expresiile de mai sus (exercițiu).

Exerciții.

a) Se dau următoarele formule ale $\mathcal{L}_p$:

$\alpha_1 : (p \supset q) \supset [(p \vee r) \supset (q \vee r)]$

$\alpha_2 : [(p \wedge q)/r] \supset (p + r)$

$\alpha_3 : [(p \equiv q) \wedge r] \vee (r \downarrow \neg r)$

$\alpha_4 : (p \vee q) \supset (p \wedge q)$

$\alpha_5 : [(p + r)/\neg r] \supset p$

1. Decideți cu ajutorul procedeului matriceal asupra acestor formule.
2. Pe baza matricelor astfel obținute construiți formele normale perfecte ale lor.
3. Verificați rezultatul astfel obținut, cu ajutorul procedeului formelor normale.
4. Determinați toate consecințele şi toate premisele fiecărei formule.

b) Ce corelație există între funcțiile *h* şi h^* din 3.4 şi formele normale perfecte?

2.1.5. Teoreme fundamentale ale $\mathcal{L}_p$

Să vedem, în fine, în acest paragraf, câteva teoreme importante ale $\mathcal{L}_p$, altele decât cele menționate în paragrafele anterioare.

Să luăm mai întâi un exemplu. Formula $\alpha : p \supset (q \supset p)$ este o formulă validă a $\mathcal{L}_p$ (verificați acest lucru). Dacă în această formulă în locul variabilei *p*, în cele două ocurențe ale sale, vom pune o formulă oarecare a $\mathcal{L}_p$, fie aceasta *p/r*, iar în locul variabilei *q* vom pune $q \wedge r$, atunci vom obține următoarea formulă a $\mathcal{L}_p$: $(p/r) \supset [(q \wedge r) \supset (p/r)]$. Cum formula α este o formulă validă şi formula obținută din α, prin substituțiile menționate, este tot o

formulă validă (exercițiu). Aşadar, orice formulă *de forma* formulei α este o formulă validă. Iar acest lucru este valabil nu numai despre formula α de mai sus, ci despre *orice* formulă validă a $\mathcal{L}_p$ în care facem substituții.

Teorema substituției. Fie $\alpha(p_1,...,p_n)$ *o formulă a* $\mathcal{L}_p$ *care conține variabilele propoziționale* $p_1,...,p_n$. *Fie* $\alpha^*(p_1/\beta_1,...,p_n/\beta_n)$ *formula obținută din* $\alpha(p_1,...,p_n)$ *prin substituția variabilelor* $p_1,...,p_n$ *în toate ocurențele lor, cu formule arbitrare* $\beta_1,...,\beta_n$. *Atunci*

Dacă $\vDash \alpha$, *atunci* $\vDash \alpha^*$.

Demonstrație.

Dacă $\vDash \alpha$, atunci α ia valoarea logică adevărat pentru orice interpretare a variabilelor $p_1,...,p_n$; aşadar, valoarea logică a formulei α este independentă de asignările făcute variabilelor ei. Însă, în α^*, în locul variabilelor $p_1,...,p_n$ intră formulele $\beta_1,...,\beta_n$ care, pentru orice interpretare a variabilelor care le conțin, iau *succesiv* valorile 1 şi 0, adică valorile logice posibile ale celor *n* variabile propoziționale din α. Şi deci şi formula α^* este indiferentă la aceste valori logice ale formulelor $\beta_1,...,\beta_n$; aşadar α^* este o formulă validă.

Remarci. 1. Conversa acestei teoreme nu este în general valabilă. Formula $(p\wedge q)\supset(p\supset q)$ este o formulă validă a $\mathcal{L}_p$. Fie $\beta_1: p\wedge q$ iar $\beta_2: p\supset q$. Dacă în locul acestor formule punem, corespunzător, variabilele *r* şi *s*, atunci, prin substituția făcută, obținem formula $r\supset s$ a $\mathcal{L}_p$.

2. Enunțul teoremei substituției conține ideea substituției *simultane* a variabilelor propoziționale $p_1,...,p_n$ cu formule arbitrare $\beta_1,...,\beta_n$. Însă într-o formulă validă a $\mathcal{L}_p$ putem alege orice număr de variabile în vederea substituirii lor, nu neaparat toate. În formula α de mai sus puteam să-l substituim pe *p* cu *p/r* şi atât. *q* putea rămâne *q* (de altfel, în acest din urmă caz, puteam considera substituția redundantă a variabilei *q* cu ea însăşi).

3. Prin substituție, dintr-o formulă validă a $\mathcal{L}_p$ putem construi o infinitate de formule ale $\mathcal{L}_p$. Prin teorema substituției şi aceste formule vor fi valide, fără a fi nevoie să le testăm validitatea.

Să luăm acum următorul exemplu. Fie $\alpha_\beta: [(p\supset q)\wedge(q/\neg r)]\supset(p\supset q)$, o formulă a $\mathcal{L}_p$ care conține subformula $\beta: p\supset q$. Fie $\alpha_\gamma: [(\neg p\vee q)\wedge(q/\neg r)]\supset(p\supset q)$ formula α_β de mai sus în care în locul subformulei β am pus formula $\gamma: \neg p\vee q$. Atunci, cum β este echivalentă cu γ, rezultă că $\alpha_\beta\equiv\alpha_\gamma$ (exercițiu). Aceasta este teorema înlocuirii, pe care am utilizat-o în câteva rânduri în paragrafele anterioare [11], chiar dacă n-am menționat-o.

Teorema înlocuirii (echivalenților). Dacă β *este o subformulă a lui* α *(simbolic* α_β*) iar* α_γ *se obține din înlocuirea a zero sau mai multe ocurențe ale subformulei* β *în* α *cu o formulă* γ, *atunci*

Dacă $\vDash\beta\equiv\gamma$, *atunci* $\vDash\alpha_\beta\equiv\alpha_\gamma$.

Demonstrație (inducție pe gradul formulei α).

[11] Comp. Formele normale în $\mathcal{L}_p$.

În demonstrația acestei teoreme ne limităm la cazul în care în α înlocuim o singură ocurențe a subformulei β cu formula γ. În felul acesta nu se restrânge nicidecum generalitatea teoremei, deoarece, evident, în cazul în care înlocuim zero ocurențe ale subformulei β cu γ, atunci α_γ este tocmai formula α, iar dacă vrem să înlocuim mai multe ocurența ale lui β cu γ în formula α, atunci repetăm aplicarea teoremei în varianta pe care-o demonstrăm.

1. α nu conține nici un operator. α este aşadar o variabilă propozițională, *p*, subformula β este tot *p*, iar α_γ este β. Cu ipoteza din enunțul teoremei (i.e. $\vDash \beta \equiv \gamma$), teorema înlocuirii are loc.

2. Presupunem că teorema are loc pentru orice formulă α al cărei grad este mai mic decât *n* $(n \geq 1)$ şi arătăm, în pasul inductiv, că teorema are loc şi pentru o formulă α al cărei grad este *n*.

În acord cu definiția conceptului de formulă, α are, în acest caz, una din următoarele forme: a) $\neg\delta$; b) $\delta \circ \varepsilon$, unde „o" denotă orice operator binar menționat în definiție.

În toate aceste cazuri gradul formulelor δ şi ε este mai mic decât *n* şi potrivit ipotezei inducției, teorema are loc pentru formulele δ şi ε; aşadar

Dacă $\vDash \beta \equiv \gamma$, atunci 1. $\vDash \delta_\beta \equiv \delta_\gamma$ şi 2. $\vDash \varepsilon_\beta \equiv \varepsilon_\gamma$

În cazul a) α are forma $\neg\delta$. Şi deci dacă $\vDash \beta \equiv \gamma$, atunci $\vDash \delta_\beta \equiv \delta_\gamma$ (prin ipoteza inducției). Însă $\vDash (\delta_\beta \equiv \delta_\gamma) \equiv (\neg\delta_\beta \equiv \neg\delta_\gamma)$. Şi astfel, dacă $\vDash \beta \equiv \gamma$, atunci $\vDash \neg\delta_\beta \equiv \neg\delta_\gamma$, adică $\vDash \alpha_\beta \equiv \alpha_\gamma$.

În cazul b) α are forma $\delta \circ \varepsilon$. Prin ipoteza inducției teorema are loc pentru δ şi ε. Şi astfel obținem $(\delta_\beta \circ \varepsilon_\beta) \equiv (\delta_\gamma \circ \varepsilon_\gamma)$ (prin Remarcă § 4.3), unde $\delta_\beta \circ \varepsilon_\beta$ este $(\delta \circ \varepsilon)_\beta$, iar $\delta_\gamma \circ \varepsilon_\gamma$ este $(\delta \circ \varepsilon)_\gamma$.

Remarcă. Între substituție şi înlocuire există următoarele deosebiri:

a) Operația substituției înseamnă substituția unei / unor *variabile propoziționale* cu formule arbitrare ale $\mathcal{L}_\text{p}$ (nu substituim formule arbitrare cu formule arbitrare!). În cazul înlocuirii, în cazul unei subformule din formula dată punem o formulă arbitrară a $\mathcal{L}_\text{p}$ (sub asumpția echivalenței lor).

b) Substituția unei variabile propoziționale *p* cu o formulă arbitrară a $\mathcal{L}_\text{p}$ trebuie realizată în *toate* ocurențele variabilei *p*; înlocuirea, în schimb, se poate realiza numeric arbitrar.

Teorema de normalitate. Pentru $n \geq 1$: $\alpha_1,...,\alpha_{n-1},\alpha_n \vDash \beta$ ddacă $\alpha_1,...,\alpha_{n-1} \vDash \alpha_n \supset \beta$.

Demonstrație (reductio ad absurdum)

a) Dacă $\alpha_1,...,\alpha_{n-1},\alpha_n \vDash \beta$ atunci $\alpha_1,...,\alpha_{n-1} \vDash \alpha_n \supset \beta$.

Presupunem 1. $\alpha_1,...,\alpha_n \vDash \beta$ şi 2 *non* $\alpha_1,...,\alpha_{n-1} \vDash \alpha_n \supset \beta$. Din 1 deducem că pentru orice interpretare a variabilelor propoziționale care apar în $\alpha_1,...,\alpha_n$, β are loc: 3. dacă $\alpha_1 \equiv ... \equiv \alpha_n = 1$, atunci $\beta = 1$ (prin definiție). Din 2 deducem că există o interpretare a variabilelor formulelor date astfel că 4. $\alpha_1 \equiv ... \equiv \alpha_{n-1} = 1$ şi 5. $\alpha_n \supset \beta = 0$. Din 5 deducem 6. $\alpha_n = 1$ şi 7. $\beta = 0$. Însă 7 contrazice 3, căci dacă $\alpha_n = 1$, atunci $\alpha_1 \equiv ... \equiv \alpha_n = 1$ şi deci $\beta = 1$.

b) Dacă $\alpha_1,...,\alpha_{n-1} \vDash \alpha_n \supset \beta$, atunci $\alpha_1,...,\alpha_n \vDash \beta$.

Presupunem 1. $\alpha_1,...,\alpha_{n-1} \models \alpha_n \supset \beta$ şi 2. *non* $\alpha_1,...,\alpha_n \models \beta$. Din 1 deducem că în orice interpretare a variabilelor din formulele date avem: 3. dacă $\alpha_1 \equiv ... \equiv \alpha_{n-1} = 1$ atunci $\alpha_n \supset \beta = 1$. Iar din 2 deducem că există o interpretare astfel că 4. $\alpha_1 \equiv ... \equiv \alpha_n = 1$ şi 5. $\beta = 0$. Însă, sub asumpția adevărului formulelor $\alpha_1,...,\alpha_{n-1}$ (din 3) am dedus $\alpha_n \supset \beta = 1$. Cum $\alpha_n = 1$ (din 4) rezultă că $\beta = 1$, ceea ce contrazice 5.

Corolar. Pentru $n \geq 1$: $\alpha_1,...,\alpha_{n-1},\alpha_n \models \beta$ ddacă $\alpha_1 \supset (...\alpha_{n-1} \supset (\alpha_n \supset \beta)...)$.

Demonstrație (prin *n* aplicări succesive ale teoremei de normalitate). Semnificația acestui corolar rezidă în aceea că problema referitoare la consecințele semantice ale unei mulțimi de formule se reduce la problema *ce formule sunt valide*.

Teorema (*). Fie $\alpha(p_1,...,p_n)$ *o formulă a* $\mathcal{L}_p$*, care conține variabilele propoziționale* $p_1,...,p_n$ *negate sau nenegate. Fie* α^* *formula care rezultă din* α *prin înlocuirea operatorilor formulei cu dualii corespunzători şi a variabilelor propoziționale* $p_1,...,p_n$ *cu negațiile lor. Atunci,* $\models \neg\alpha \equiv \alpha^*$ (echivalent: $\models \alpha \equiv \neg\alpha^*$).

Demonstrație.

Este suficient să demonstrăm teorema pentru cazul în care singurii operatori binari sunt $\wedge$ şi $\vee$, căci utilizând $\neg$ orice formulă a $\mathcal{L}_p$ poate fi exprimată printr-o formulă care conține doar aceşti trei operatori (cf. Th. 2, 2.1.3.4). O dată adusă într-o asemenea formă, pe baza legilor lui De Morgan (15, 27)[12] negația unei formule care conține operatori binari se deplasează succesiv în interiorul formulei, permutând reciproc operatori binari ai formulei. Utilizând legile eliminării negațiilor multiple (1, 2) simplificăm formula astfel încât orice variabilă propozițională a formulei va fi negată cel mult o dată.

Exemplu.

Fie $\alpha : (p \equiv q) \downarrow \neg r$. Atunci $\alpha^* : (\neg p + \neg q) / \neg\neg r$.

Transformăm formula α într-o formulă echiveridică (şi deci echivalentă) care conține doar operatorii $\wedge, \vee$ şi $\neg$.

$\alpha : (p \equiv q) \downarrow \neg r$ eq $\{[(p \wedge q) \vee (\neg p \wedge \neg q)] \vee \neg r\}$ eq $\neg[(p \wedge q) \vee (\neg p \wedge \neg q)] \wedge \neg\neg r$ eq $\neg[(p \wedge q) \wedge \neg(\neg p \wedge \neg q)] \wedge \neg\neg r$ eq $(\neg p \vee \neg q) \wedge (\neg\neg p \vee \neg\neg q) \wedge \neg\neg r$ eq
eq $(\neg p \vee \neg q) \wedge (p \vee q) \wedge r$.

La fel procedăm cu α^*.

$\alpha^* : (\neg p + \neg q) / \neg\neg r$ eq $\neg\{[(\neg p \wedge \neg\neg q) \vee (\neg\neg p \wedge \neg q)] \wedge \neg\neg r\}$ eq
$\neg[(\neg p \wedge \neg\neg q) \vee (\neg\neg p \wedge \neg q)] \vee \neg\neg\neg r$ eq $[\neg(\neg p \wedge \neg\neg q) \wedge \neg(\neg\neg p \wedge \neg q)] \vee \neg\neg\neg r$ eq
$[(\neg\neg p \vee \neg\neg\neg q) \wedge (\neg\neg\neg p \vee \neg\neg q)] \vee \neg\neg\neg r$ eq $[(p \vee \neg q) \wedge (\neg p \vee q)] \vee \neg r$

Această din urmă formulă poate fi mai departe transformată echiveridic, astfel încât negând formula α, de mai sus, să obținem exact formula α^*. Aşadar,

[12] Cifrele indică formulele corespunzătoare din lista paragrafului 2.1.2.3.

$[(p \vee \neg q) \wedge (\neg p \vee q)] \vee \neg r$ eq $(p \wedge \neg p) \vee (p \wedge q) \vee (\neg p \wedge \neg q) \vee (q \wedge \neg q) \vee \neg r$ eq
$(p \wedge q) \vee (\neg p \wedge \neg q) \vee \neg r$.

Acum, dacă negăm formula α obţinem:
$\neg\alpha : \neg[(\neg p \vee \neg q) \wedge (p \vee q) \wedge r]$ eq $\neg\{[(\neg p \vee \neg q) \wedge (p \vee q) \wedge r]\}$ (prin 5) eq
eq $\neg[(\neg p \vee \neg q) \wedge (p \vee q)] \vee \neg r$ eq $\neg(\neg p \vee \neg q) \vee \neg(p \vee q) \vee \neg r$ eq
eq $(\neg\neg p \wedge \neg\neg q) \vee (\neg p \wedge \neg q) \vee \neg r$ eq $(p \wedge q) \vee (\neg p \wedge q) \vee \neg r$ eq α^*.

Teorema dualităţii.

Fie α, β *formule ale* $\mathcal{L}_p$ *în care apar variabilele propoziţionale* $p_1, ..., p_n$, *negate sau nenegate. Fie* α^δ *şi* β^δ *formulele care rezultă din* α, *respectiv* β, *înlocuind operatorii logici cu dualii corespunzători. Atunci*:

1. *Dacă* $\vDash \alpha$, *atunci* $\vDash \neg\alpha^\delta$.
2. *Dacă* $\vDash \neg\alpha$, *atunci* $\vDash \alpha^\delta$.
3. *Dacă* $\vDash \alpha \supset \beta$, *atunci* $\vDash \beta^\delta \supset \alpha^\delta$.
4. *Dacă* $\vDash \alpha \equiv \beta$, *atunci* $\vDash \alpha^\delta \equiv \beta^\delta$.

Demonstraţie. Pentru considerentele menţionate în demonstraţia teoremei precedente, vom considera că singurii operatori binari ai formulelor α, β sunt $\wedge$ şi $\vee$.

1. Presupunem $\vDash \alpha$. Prin Teorema (*) şi exerciţiul 3b din 2.1.2.4 rezultă $\vDash \neg\alpha^*$. Însă dacă în această formulă validă, $\neg\alpha^*$, substituim simultan variabilele propoziţionale $p_1, ..., p_n$ cu negaţiile lor, atunci obţinem formula $\neg\alpha^{**}$, de asemenea validă (prin teorema substituţiei). Iar dacă în această formulă eliminăm negaţiile multiple, obţinem formula validă $\neg\alpha^\delta$. Aşadar, $\vDash \neg\alpha^\delta$.

Exemplu. Fie $\vDash \alpha : (p \wedge \neg q) \vee [(q \wedge \neg r) \vee (\neg p \vee r)]$; atunci
$\vDash \neg\alpha^* : \neg\{(\neg p \vee \neg\neg q) \wedge [(\neg q \vee \neg\neg r) \wedge (\neg\neg p \wedge \neg r)]\}$; prin Th (*) şi exerc. 3b (paragr. 2.1.2.4).
$\vDash \neg\alpha^{**} : \neg\{(\neg\neg p \vee \neg\neg\neg q) \wedge [(\neg\neg q \vee \neg\neg\neg r) \wedge (\neg\neg\neg p \wedge \neg\neg r)]\}$; prin Th substituţiei
$\vDash \neg\alpha^\delta : \neg\{(p \vee \neg q) \wedge [(q \vee \neg r) \wedge (\neg p \wedge r)]\}$; prin eliminarea negaţiilor multiple
(Verificarea validităţii formulei $\neg\alpha^\delta$: exerciţiu).

2. (similar).

3. Presupunem antecedentul condiţionalului $\vDash \alpha \supset \beta$. Şi astfel $\vDash \neg\beta \supset \neg\alpha$ (prin contrapoziţia implicaţiei). Şi deci $\vDash \beta^* \supset \alpha^*$ (căci dacă în formulele $\neg\beta$ şi $\neg\alpha$ înlocuim operatorii cu dualii corespunzători şi variabilele propoziţionale cu negaţiile lor, atunci obţinem, prin Th (*), formulele echivalente $\neg\neg\beta^*$ şi $\neg\neg\alpha^*$, adică β^* şi α^*; de unde, prin exerc 3b, obţinem $\vDash \beta^* \supset \alpha^*$). Şi astfel $\vDash \beta^{**} \supset \alpha^{**}$ (prin th. substituţiei) şi deci $\vDash \beta^\delta \supset \alpha^\delta$ (eliminarea negaţiilor multiple).

4. (similar).

Exerciții.

1. Construiți, pe baza teoremei (*), α^* corespunzătoare următoarelor formule:
$\alpha_1 : [(p \not\subset q)/ r] \wedge \neg r$
$\alpha_2 : [(p + q) \vee \neg r] \equiv (p \wedge q)$
$\alpha_3 : [(p \downarrow q) \vee r] \supset [(p \wedge q)/ \neg r]$
$\alpha_4 : [(p \wedge q) \supset r] \not\subset \neg r$

2. Arătați că $\vDash \neg\alpha \equiv \alpha^*$ în fiecare caz în parte.

3. Fie $\alpha : [(p \wedge \neg q) \supset (r \vee \neg r)]/(p \wedge \neg p)$. Aplicați teorema dualității formulei α şi arătați, cu ajutorul procedeului matriceal că relația condițională: dacă $\vDash \alpha$, atunci $\vDash \neg\alpha^{\delta}$ are loc.